

**Vysoká škola báňská – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra elektroniky**



**TESTER ELEKTRICKÝCH MOTORKŮ PRO  
NASTAVOVÁNÍ POLOHY SVĚTLOMETŮ  
AUTOMOBILŮ**

**TESTER OF ELECTRIC MOTORS FOR VEHICLE'S  
HEADLIGHT LEVELING SYSTEM**

**2020**

**Bc. Marek Kubatko**

## Zadání diplomové práce

Student: **Bc. Marek Kubatko**  
Studijní program: N2649 Elektrotechnika  
Studijní obor: 2612T003 Aplikovaná elektronika  
Téma: **Tester elektrických motorků pro systém nastavování polohy světlometů automobilu**  
**Tester of Electric Motors for Vehicle's Headlight Leveling System**

Jazyk vypracování: čeština

### Zásady pro vypracování:

1. Proveďte teoretickou rešerši prostředků pro nastavování předních světlometů vozidel.
2. Navrhněte testovací stanoviště pro testování vybraných typů elektrických motorků.
3. Realizujte testovací zařízení s měřením polohy nastavovací hřídele motorku.
4. Prakticky ověřte správnou funkci testeru včetně porovnání získaných výsledků.

### Seznam doporučené odborné literatury:

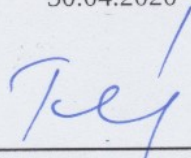
JURGEN, R. K.: Automotive electronics handbook. 2nd ed. New York: McGraw-Hill, c1999. McGraw-Hill handbooks. ISBN 0-07-034453-1.  
WÖRDENWEBER, B., WALLASCHEK, J., BOYCE, P., HOFMANN, D.D.: Automotive Lighting and Human Vision, Springer Berlin Heidelberg, 2009, ISBN 978-3-5408-2694-1.  
NEBORÁK, I., SLÁDEČEK, V.: Elektrické pohony. Učební texty pro kombinované a distanční studium. VŠB-TU Ostrava, 2017.  
BOLDEA, I.-NASAR, S. A.: Electric Drives. Third Edition, CRC Press, 2016. ISBN 978-1-4987-4820-9.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

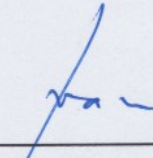
Vedoucí diplomové práce: **Ing. Tomáš Mrověk, Ph.D.**

Datum zadání: 01.09.2019

Datum odevzdání: 30.04.2020

  
doc. Ing. Petr Palacký, Ph.D.  
vedoucí katedry



  
prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 15. května 2020

  
.....

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

Dne: 15. května 2020

  
.....

## **Poděkování**

Rád bych poděkoval Ing. Tomáši Mrověcovi Ph.D., Ing. Martinu Sobkovi, Ph.D., Ing. Danielu Kouřilovi, Ing. Janu Michalíkovi, Ing. Martinu Nevřelovi, Tomáši Segedřovi, Janu Sznapkovi a Bc. Radku Petrášovi za odbornou pomoc a konzultaci při vytváření této diplomové práce.

## **Abstrakt**

Tato práce se v teoretické části zabývá osvětlovacími systémy automobilů, detailním popisem sběrnice LIN a stručným rozbořem analyzovaných elektrických motorů. Praktická část je zaměřena na návrh, realizaci a ověření testovacího přípravku pro kontrolu elektrických motorů pro nastavování úrovně osvětlení vozovky ve společnosti zabývající se výrobou automobilového osvětlení. Speciálně pro tento účel byl navržen testovací box, navrhnout a vyroben napájecí zdroj a naprogramován řídicí software v jazyce Wiring, typickým pro vývojovou platformu Arduino. Výsledkem práce je přenosné testovací stanoviště pro analýzu elektrických motorů.

## **Klíčová slova**

LIN sběrnice; Krokový motor; Stejnosměrný motor; Arduino; Testovací přípravek; Vector CANoe; Napájecí zdroj

## **Abstract**

In the theoretical part, this thesis deals with car lighting systems, a detailed description of the LIN bus and a brief analysis of the electric motors. The practical part is focused on the design, implementation and verification of tester of electric motors for setting the level of road lighting in a company engaged in the production of automotive lighting. The box for tester was designed specifically for this purpose, the power supply was made and designed, and the control software in the Wiring language, typical for the Arduino development platform, was programmed. The result of the thesis is a portable tester for the analysis of electric motors.

## **Key words**

LIN bus; Stepper motor; DC motor; Arduino; Tester; Vector CANoe; Power supply

# Obsah

Seznam použitých zkratk a symbolů .....	- 10 -
Seznam obrázků .....	- 12 -
Seznam tabulek .....	- 14 -
Úvod.....	- 15 -
1 Prostředky pro nastavování předních světlometů vozidel .....	- 16 -
1.1 Osvětlovací automobilové systémy .....	- 16 -
1.2 Řídící elektronika světlometu.....	- 17 -
1.3 LIN bus.....	- 19 -
1.3.1 Princip komunikace sběrnice.....	- 19 -
1.3.2 Hardwarové řešení sběrnice .....	- 22 -
1.3.3 Komunikační protokol.....	- 23 -
1.4 Krokové motory .....	- 24 -
1.5 Stejnosměrné motory.....	- 26 -
2 Návrh testovacího stanoviště pro kontrolu elektrických motorů.....	- 28 -
2.1 Návrh napájecího zdroje.....	- 28 -
2.1.1 Elektrické schéma a dimenzace jednotlivých prvků zdroje.....	- 28 -
2.1.2 Návrh 3D modelu napájecího zdroje.....	- 31 -
2.2 Návrh testovací části přípravku pro krokový motor.....	- 32 -
2.2.1 Řídící část krokového motoru .....	- 32 -
2.2.2 Budič sběrnice .....	- 33 -
2.2.3 Snímač proudu.....	- 35 -
2.2.4 Snímač vysunutí jezdce motorku.....	- 35 -
2.2.5 Zobrazovací jednotka .....	- 36 -
2.3 Návrh testovací části přípravku pro stejnosměrný motor.....	- 37 -
2.4 Řídící část stejnosměrného motoru .....	- 38 -
3 Realizace testovacího zařízení .....	- 39 -
3.1 Napájecí zdroj testeru elektrických motorků.....	- 41 -
3.2 Řídící software .....	- 42 -
3.3 Princip komunikace s krokovým motorkem po sběrnici LIN .....	- 44 -



3.3.1	Dynamické adresování .....	- 46 -
3.3.2	Nastavení parametrů .....	- 46 -
3.3.3	Vyčtení parametrů .....	- 47 -
3.3.4	Zjištění aktuální polohy pomocí dynamických adres .....	- 49 -
3.4	Vector CANoe .....	- 50 -
3.4.1	CANoe.LIN .....	- 50 -
3.4.2	Převodník VN8914A .....	- 51 -
4	Ověření funkčnosti testeru .....	- 53 -
4.1	Ověření LIN komunikace .....	- 53 -
4.2	Ověření měření polohy motorku .....	- 53 -
4.3	Ověření testovací funkce Auto .....	- 54 -
4.4	Ověření testovací funkce Manual .....	- 56 -
4.5	Oživení a otestování napájecího zdroje .....	- 57 -
Závěr .....		- 59 -
Použitá literatura .....		- 61 -

## Seznam použitých zkratek a symbolů

Zkratka	Význam
<b>LED</b>	Light emitting diode
<b>LIN</b>	Local interconnect network
<b>USB</b>	Universal serial bus
<b>C++</b>	Programovací jazyk
<b>CAN</b>	Controller area network
<b>DRL</b>	Daytime Running Lights
<b>SBL</b>	Static bending light
<b>UART</b>	Universal asynchronous receiver-transmitter
<b>PCB</b>	Printed circuit board
<b>CAD</b>	Computer aided design
<b>SDA</b>	Serial data
<b>SCL</b>	Serial clock
<b>Rx</b>	Receive
<b>Tx</b>	Transmission
<b>KM</b>	Krokový Motor
<b>DCM</b>	Stejnoseměrný motor (Direct current)
<b>UI</b>	User interface
<b>ESD</b>	Electrostatic discharge
<b>LCD</b>	Liquid Crystal Display
<b>I2C</b>	Komunikační sběrnice
<b>PWM</b>	Pulse width modulation
<b>ECU</b>	Engine Control Unit
$V_{BAT}$	Napětí baterie [V]
$U_{cc}$	Napájecí napětí [V]
$U_R$	Řídící napětí [V]
$U_{12RECT}$	Úbytek napětí na diodovém můstku (12 V větev) [V]

---

$U_{12vst}$	Vstupní napětí na diodovém můstku (12 V větev) [V]
$U_f$	Úbytek napětí na diodě v propustném směru [V]
$P_{12RECT}$	Výkonová ztráta diodového můstku (12 V větev) [W]
$I_{12max}$	Maximální možný proud (12 V větev) [A]
$U_{05RECT}$	Úbytek napětí na diodovém můstku (5 V větev) [V]
$P_{05RECT}$	Výkonová ztráta diodového můstku (5 V větev) [W]
$U_{05vst}$	Vstupní napětí na diodovém můstku (5 V větev) [V]
$I_{05max}$	Maximální možný proud (5 V větev) [A]
$C_{12}$	Kapacita filtračního kondenzátoru (12 V větev) [F]
$U_{12min}$	Minimální napětí na vstupu stabilizátoru (12 V větev) [V]
$U_{12výst}$	Výstupní napětí na stabilizátoru 7812 [V]
$C_{05}$	Kapacita filtračního kondenzátoru (5 V větev) [F]
$U_{05min}$	Minimální napětí na vstupu stabilizátoru (5 V větev) [V]
$U_{05výst}$	Výstupní napětí na stabilizátoru 7805 [V]
$P_{12}$	Výkonová ztráta stabilizátoru 7812 [W]
$R_{12thj-c}$	Tepelný odpor přechod – pouzdro [K/W]
$T_j$	Teplota přechodu [K]
$T_{12c}$	Teplota pouzdra (12 V větev) [K]
$R_{12thc-a}$	Tepelný odpor chladič – okolí (12 V větev) [K/W]
$P_{05}$	Výkonová ztráta stabilizátoru 7805 [W]
$R_{05thj-c}$	Tepelný odpor přechod – pouzdro [K/W]
$T_j$	Teplota přechodu [K]
$T_{05c}$	Teplota pouzdra (5 V větev) [K]
$R_{05thc-a}$	Tepelný odpor chladič – okolí (5 V větev) [K/W]
$R_i$	Vnitřní odpor zdroje [ $\Omega$ ]
$Bd$	Baud Rate [bit/s]
$f$	Frekvence [Hz]

---

## Seznam obrázků

Obrázek 1.1:	Asymetrické tlumené světlo [2] .....	- 16 -
Obrázek 1.2:	Automatický levelingový systém [3] .....	- 17 -
Obrázek 1.3:	Blokové schéma struktury řídicí elektroniky světlometu Land Rover .....	- 17 -
Obrázek 1.4:	Příklad komunikační struktury master a slave uzlů na sběrnici .....	- 19 -
Obrázek 1.5:	Znázornění komunikace master, slave .....	- 20 -
Obrázek 1.6:	Znázornění komunikace slave, master .....	- 20 -
Obrázek 1.7:	Znázornění komunikace slave, slave.....	- 21 -
Obrázek 1.8:	Komunikace na sběrnici mezi Master jednotkou a slave jednotkami .....	- 21 -
Obrázek 1.9:	Znázornění napěťových úrovní pro vysílač a přijímač.....	- 22 -
Obrázek 1.10:	Obecné elektrické schéma budiče sběrnice[6] .....	- 23 -
Obrázek 1.11:	Příklad rámce zprávy komunikačního protokolu .....	- 24 -
Obrázek 1.12:	Levelingový krokový motor pro nastavení vertikálního sklonu.....	- 25 -
Obrázek 1.13:	Bendingový krokový motor pro nastavení horizontálního přisvětlení .....	- 25 -
Obrázek 1.14:	Levelingový stejnosměrný motor pro nastavení vertikálního sklonu.....	- 26 -
Obrázek 1.15:	Funkční charakteristika vysunutí hřídele motorku – poměr $U_{cc}/U_{ř}$ .....	- 27 -
Obrázek 1.16:	Blokové schéma síťového napájecího zdroje .....	- 28 -
Obrázek 1.17:	Elektrické schéma napájecího zdroje pro tester motorků.....	- 29 -
Obrázek 1.18:	Obrázek použitého chladiče [11].....	- 31 -
Obrázek 1.19:	3D model napájecího zdroje přípravku .....	- 31 -
Obrázek 1.20:	Blokové schéma části přípravku pro krokový motor .....	- 32 -
Obrázek 1.21:	Arduino ATmega 2560[12] .....	- 33 -
Obrázek 1.22:	Elektrické schéma zapojení budiče TJA1028T .....	- 34 -
Obrázek 1.23:	Rozložení součástek dps budiče TJA1028 .....	- 34 -
Obrázek 1.24:	Snímač proudu ACS712 [15] .....	- 35 -
Obrázek 1.25:	Posuvné měřítko s displejem a digitálními vstupy [16] .....	- 35 -
Obrázek 1.26:	Měnič hladiny napětí mezi posuvným měřítkem a Arduinem .....	- 36 -
Obrázek 1.27:	Použitý LCD displej s rozlišením 16x2 segmentů [17].....	- 36 -
Obrázek 1.28:	Převodník k displeji PCF8574 [18] .....	- 37 -
Obrázek 1.29:	Blokové schéma části přípravku pro stejnosměrný motor .....	- 37 -
Obrázek 1.30:	Měnič hladiny napětí mezi Arduinem a řídicí elektrodou DC motoru.....	- 38 -

Obrázek 1.31:	Konstrukční část přípravku pro krokový motorek.....	39 -
Obrázek 1.32:	Konstrukční část přípravku pro stejnosměrný motor .....	40 -
Obrázek 1.33:	3D model a finální verze testovacího přípravku.....	40 -
Obrázek 1.34:	Rozložení součástek dps napájecího zdroje v programu EAGLE.....	41 -
Obrázek 1.35:	Vyhotovený napájecí zdroj přípravku .....	41 -
Obrázek 1.36:	Průběh programu .....	42 -
Obrázek 1.37:	Diagram testovací sekvence autotest.....	43 -
Obrázek 1.38:	Diagram testovací sekvence Manual test .....	44 -
Obrázek 1.39:	Trace window zpráv odposlechnutých na LIN sběrnici.....	51 -
Obrázek 1.40:	Ukázka rozšíření LDF explorer[22] .....	51 -
Obrázek 1.41:	Blokové schéma zapojení převodníku [24] .....	52 -
Obrázek 1.42:	Záznam komunikace mezi krokovým motorem a řídicí jednotkou.....	53 -
Obrázek 1.43:	Data z posuvného měřítka SDA (horní průběh) a SCL (dolní průběh) .....	54 -
Obrázek 1.44:	Komunikace mezi KM a řídicí jednotkou při režimu manual .....	56 -
Obrázek 1.45:	Zleva: průběh výstupního napětí ze stabilizátoru 7812 (fialový) 7805 (modrý) a průběh výkonové ztráty na stabilizátorech 7812 (zelený) a 7805 (červený) .....	57 -
Obrázek 1.46:	Závislost výstupního napětí stabilizátoru 7805 na proudu protékajícím elektronickou zátěží .....	58 -
Obrázek 1.47:	Závislost výstupního napětí stabilizátoru 7812 na proudu protékajícím elektronickou zátěží .....	58 -

## Seznam tabulek

Tabulka 1.1:	Princip komunikace s motorkem pomocí statických adres .....	- 45 -
Tabulka 1.2:	Princip komunikace s motorkem pomocí dynamických adres .....	- 45 -
Tabulka 1.3:	Nastavování a zjišťování polohy pomocí statických adres .....	- 45 -
Tabulka 1.4:	Nastavování a zjišťování polohy pomocí dynamických adres .....	- 45 -
Tabulka 1.5:	Rozpis dat příkazu Dynamic assist ID .....	- 46 -
Tabulka 1.6:	Rozpis dat příkazu SetMotorParam.....	- 47 -
Tabulka 1.7:	Rozpis dat příkazu GetFullStatus – preparing frame .....	- 47 -
Tabulka 1.8:	Rozpis dat příkazu GetFullStatus – Response frame 1 .....	- 48 -
Tabulka 1.9:	Rozpis dat příkazu GetFullStatus – Response frame 2 .....	- 48 -
Tabulka 1.10:	Rozpis dat příkazu GetActPos – Direct ID verze .....	- 49 -
Tabulka 1.11:	Legenda k přiřazení dynamických adres .....	- 49 -
Tabulka 1.12:	Hodnoty měření pro kontrolu posuvného měřítka.....	- 54 -
Tabulka 1.13:	Odečtené data ze sériového portu režimu autotest DC – motor OK .....	- 55 -
Tabulka 1.14:	Odečtené data ze sériového portu režimu autotest DC – motor NOK.....	- 55 -
Tabulka 1.15:	Odečtené data ze sériového portu režimu autotest KM – motor OK.....	- 55 -
Tabulka 1.16:	Odečtené data ze sériového portu režimu autotest KM – motor NOK.....	- 55 -

## Úvod

V dnešní době, kdy se elektronika v automobilech vyskytuje stále ve větší míře, než tomu bylo v předchozích letech, roste i náročnost nejen na údržbu, ale také na samotný vývoj a výběr použitých technologií pro zvolené aplikace. Žárovkové světlomety nahradily LED světlomety, které mají oproti svým předchůdcům mnohem vyšší účinnost, ale jsou o to náročnější na řídicí algoritmy. Veškeré komfortní systémy od nastavování volantů, sedadel, ovládání oken, klimatizace a regulátoru rychlosti jízdy přes zabezpečovací systémy s centrálním zamykáním, imobilizérem až po zmíněné osvětlovací systémy nebo informační a navigační systémy. Všechny tyto zmíněné systémy včetně snímačů a akčních členů mají jednu společnou vlastnost, a to že mezi sebou komunikují po speciální datové sběrnici.

Tato práce se zaměřuje na jeden určitý akční člen používaný pro nastavování výšky směřovaného světelného toku čelních světel v automobilech, jako jsou Jaguar, Bentley, Land Rover, Volkswagen, Ford a Seat. Pomocí speciálních elektrických motorků řízených řídicí jednotkou umístěnou v každém světle, tato řídicí jednotka nastavuje úroveň osvětlení vozovky, tak aby řidič viděl dostatečně daleko, ale aby zároveň nedocházelo k oslňování protijedoucích vozidel.

Hlavním úkolem práce bylo navrhnout přípravek určený k testování těchto motorků, které se používají pro zmíněný účel. Mikrokontrolér v přípravku ve své funkci nahrazuje řídicí jednotku světlometu pro zajištění stejných podmínek jako při normálním provozu a komunikuje s motorkem po datové sběrnici LIN. Tuto komunikaci je možno monitorovat převodníkem LIN to USB VN8914A od firmy Vector připojenému k osobnímu počítači. Je zde také možnost řídit přípravek jednoduchým ovládáním na čelním panelu.

Program pro mikrokontrolér je napsán v jazyce Wiring, který je nadstavbou pro jazyk C++. Tento jazyk je typický pro použitý jednočipový mikrokontrolér vývojové platformy Arduino. Pro komunikaci po sběrnici LIN byla použita speciální knihovna, která používá komunikační protokol verze 1.3 pro komunikaci mezi mikrokontrolérem a řadičem motoru.

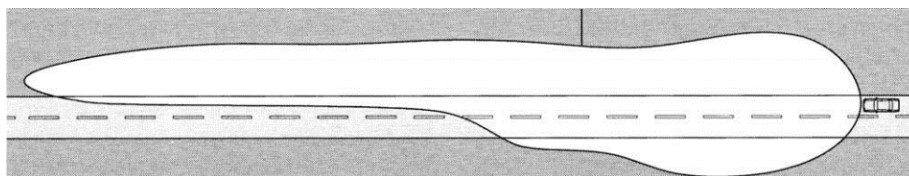
Přínosem této práce je především možnost využití přípravku při výrobním procesu společnosti zabývající se výrobou moderních světlometů. Přípravek usnadňuje a urychluje analýzu jednotlivých motorků a zefektivňuje identifikaci vadných kusů.

# 1 Prostředky pro nastavování předních světlometů vozidel

Při vývoji automobilů jsou kladeny nároky na snižování spotřeby a zvyšování bezpečnosti v každodenním provozu. Testování světlometů je klíčovou součástí při jejich výrobě, jednotlivé komponenty jsou testovány odděleně a následně i jako zkompleťovaný celek. Jednou z částí testování je právě nastavování polohy předních světlometů. Pohyb světlometu ve vodorovné i horizontální poloze zajišťuje tak zvaný inteligentní krokový motor, který je tvořen bipolárním krokovým motorem s výkonovou elektronikou. Jádrem krokového motoru je integrovaný obvod, řadič, který implementuje řízení krokového motoru, diagnostiku a rozhraní pro komunikaci se systémem s vyšší úrovní prostřednictvím speciální sběrnice. Mezi hlavní výhody tohoto krokového motoru patří mikrokrokování, tichý chod, diagnostika a velká odolnost vůči elektromagnetickému rušení. Jako levnější a méně přesná varianta za krokový motor se používá motor stejnosměrný, který se liší od krokového absencí datové sběrnice, jelikož pro jeho řízení je zde pouze klíčová hladina napětí vysílaná řídicí jednotkou, vztahenou k hladině palubního napětí.

## 1.1 Osvětlovací automobilové systémy

Dříve používané symetrické svícení bylo v roce 1957 nahrazeno asymetrickým rozdělením světla. Asymetrické rozdělení osvětlení v úrovni vozovky se liší podle trhu, na který jsou automobily určeny. V zemích s pravostranným řízením se výrazně zvětšil dosah tlumeného světla na pravé straně automobilu, aniž by docházelo k oslnění protijedoucích vozidel. [1].



Obrázek 1.1: *Asymetrické tlumené světlo [2]*

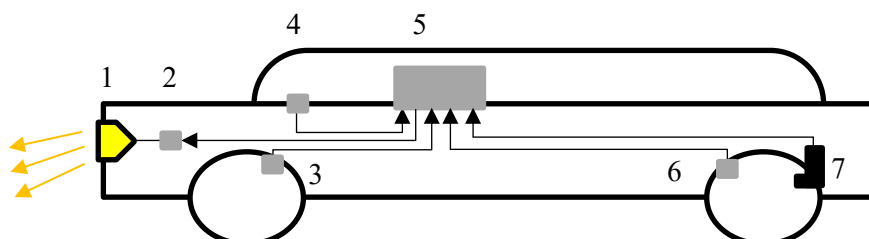
Systémy světlometů starající se nejen o intenzitu, ale také o směr vyzařovaného paprsku se dají rozdělit podle funkce regulace dosahu světlometu na manuální a automatické. Tato regulace má při všech stavech zatížení vozidla zabezpečit stálý dobrý dohled bez oslnování protijedoucích vozidel řízením úhlu sklonu tlumeného světla. Manuální nastavování je realizováno ručně řidičem pomocí spínače. Tento typ nastavení výšky osvětlení je subjektivní, a ne vždy řidič nastaví úroveň správně. Automatické nastavování sklonu světlometů je zajišťováno bez nutnosti zásahu řidiče neustále, a proto by měla být o to bezpečnější a jistější. Automatické nastavování se dále rozlišuje na statické a dynamické.

Statický systém pouze rozlišuje změnu náklonu vozidla v důsledku změny stavu zatížení a podle toho nastavuje hladinu osvětlení vozovky.

Dynamické systémy reagují jak na náklon vozidla v důsledku změny stavu zatížení, tak také na změnu sklonu související s jízdou, tím jsou myšleny stavy zejména při rozjezdech a



brždění. Ke komponentům zařízení automatické regulace patří snímače na nápravách vozidla, které snímají úhel sklonu karoserie, elektronická řídicí jednotka, která ze signálů snímačů vypočte úhel náklonu a tento úhel porovná s nastavenou hodnotou. Při odchylce vyšle příslušný signál na servomotory a ty pak provedou přesné nastavení světlometů. [1]

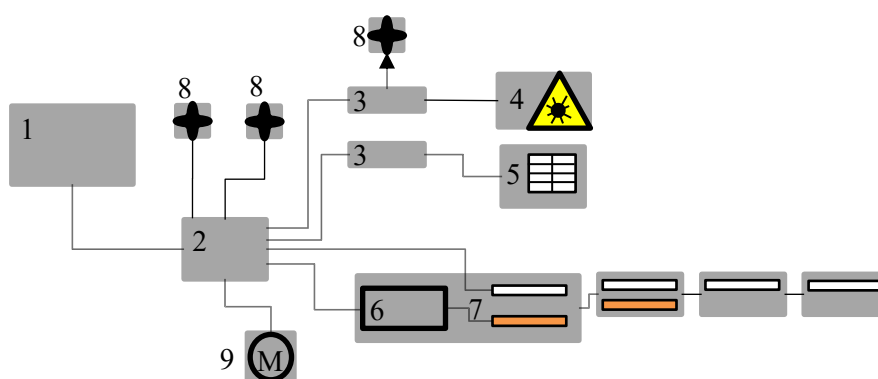


Obrázek 1.2: Automatický levelingový systém [3]

Na obrázku lze vidět komplexní řídicí systém zajišťující automatické nastavení výšky osvětlení vozovky. Všechny komponenty tohoto systému byly očíslovány pro snazší popis. 1 Lampa, 2 Akční člen, 3 Senzor náklonu přední nápravy, 4 Spínač pro nastavení úrovně osvětlení, 5 Centrální řídicí jednotka, 6 Senzor náklonu zadní nápravy, 7 Senzor pro měření otáček

### 1.2 Řídicí elektronika světlometu

Světlomet jako takový už není propojen s palubní sítí pouze napájecími vodiči, ani není propojen pouze datovými vodiči mezi centrální řídicí jednotkou a akčními členy světlometu. Světlomet má totiž vlastní řídicí jednotku, která si sama ovládá a zároveň kontroluje komunikaci s veškerými členy na výstupu, jako jsou svítidla a mechanická struktura, která ovládá směr světelného paprsku. Řídicí jednotka světlometu komunikuje s centrální řídicí jednotkou automobilu, prostřednictvím hlavní kabeláže, a na základě příkazů centrální řídicí jednotky, tyto povely zpracovává a řídí s nimi své výstupy.



Obrázek 1.3: Blokové schéma struktury řídicí elektroniky světlometu Land Rover

Popis k očíslovaným komponentám vnitřní struktury řídicí elektroniky Obrázek 1.3:  
1 - řídicí jednotka automobilu, 2 - řídicí jednotka světlometu, 3 - switchboardy pro řízení

podřízených jednotek, 4 - laserová jednotka světlometu, 5 - pixel jednotka světlometu, 6 - integrovaný obvod pro animaci blinkru, 7 - led modul směrového a denního svícení, 8 - ventilátory pro proudění vzduchu ve světlometu 9 - krokový motor

Veškerá elektronická zařízení a řídicí systémy v dopravních prostředcích, včetně světelného systému, potřebují mezi sebou komunikovat v reálném čase. K tomu musí být samozřejmě připojeny k napájecímu zdroji, nazývaným také jako palubní napájecí síť. Všechna tato komunikace probíhá prostřednictvím speciální datové sítě. Tyto systémy jsou propojeny se snímači a akčními členy, které samy zpracovávají analogové signály a starají se o jejich převod na signál elektronický. V případě, že je snímač měřené veličiny řešen jako samostatný modul, tento signál dále putuje do centrální řídicí jednotky, kde je zpracováván. Automobilová sběrnice byla navržena tak aby, komunikace mezi snímači, řídicí jednotkou a akčními členy příliš nezatěžovala centrální řídicí jednotku a byla také dostatečně robustní a odolná vůči elektromagnetickému rušení. Mezi nejčastěji používané sběrnice v automotive průmyslu řadíme:

- CAN bus
- LIN bus
- Flex Ray
- K-line, L-line

Dnešní světlomety se už neskládají pouze ze žárovek, jejichž veškeré ovládání obstarával řidič. V nejnovějších světlometech se používají výkonové LED svítidla, halogenové žárovky, xenonové výbojky, nebo také dokonce i laserové technologie. Tyto svítidla mají zajišťovat vhodné osvětlení pro jízdu v různých podmínkách jako je jízda ve městě na dálnici nebo mezi městy s mnohem větší efektivitou, než tomu bylo u jejich předchůdců, navíc dnes zákazník už také požaduje, aby svítidlo umělo hned několik funkcí. Mezi tyto funkce se řadí denní svícení označované jako DRL, tento nízkospotřebový režim se používá během dne, pokud je na vozovce dobrá viditelnost, další funkcí je SBL, toto přisvěcování do zatáček využívají automobily zejména v prudkých zatáčkách, kdy je toto přisvěcování nejvíce znatelné, aktivované je ale vždy, když se spustí směrová světla a je také zohledněna rychlost vozidla. Funkce Low beam a High beam se používají při klasickém svícení a svícení tzv. dálkovými světly. Světlomety už také zvládají tak zvanou animaci blinkru, nebo dokážou zamezit osvětlení protijedoucího vozidla. Moderní světlomet dostává všechny potřebné informace o rychlosti vozidla, poloze volantu a podle řídicích algoritmů rozsvěcuje požadované světelné funkce, a nastavuje mechanické komponenty světlometu. Řídicí elektronika světlometu pak odesílá informace do centrální řídicí jednotky o tom, zda byly požadavky splněny, nebo pokud při jejich nastavování došlo k chybě. Tyto chytré světlomety však mají díky své složité elektronice jednu nevýhodu, a to že v případě poruchy už pro jejich analýzu nestačí pouhý multimetr, ale je potřeba mít speciální diagnostické zařízení určené pro práci se zvoleným typem světla. Ve většině případech se pro diagnostiku používá počítač se speciálním softwarem s měřicí kartou pro měření analogových veličin.

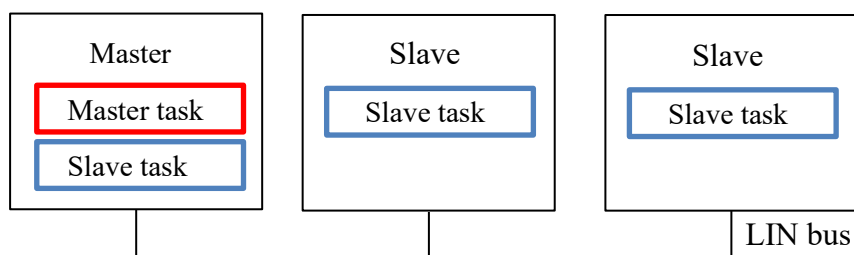
### 1.3 LIN bus

Nejrozšířenější sběrnici pro práci se světlomety je sběrnice LIN. U světlometů se sběrnice začala upřednostňovat, jelikož oproti sběrnici CAN má nižší výrobní náklady a její přenosová rychlost pro ovládání elektroniky světlometu je dostačující. Koncerny jako BMW, Volkswagen, Audi, Volvo nebo Mercedes-Benz se podílely na jejím vzniku a v roce 1999 ji uvedly na trh. Od té doby prošla několika modifikacemi a od roku 2016 se používá standart splňující normu ISO 17987:2016. Vývoj sběrnice byl od začátku směřován na snížení nákladů s implementací komunikační sběrnice. LIN nenahrazuje sběrnici CAN, ale využije se v případech, kdy není potřeba tak vysoké přenosové rychlosti, například při ovládání a polohování zrcátek, nastavování sedadel, řízení klimatizace a stěračů stahování oken, centrální zamykání dveří, a především ovládání osvětlení. Výhodou LIN sběrnice je absence potřeby používat speciální řadiče, pro řízení stačí mikropočítač s vnitřním obvodem UART a jako oscilátor stačí jednoduchý RC obvod. Právě nízká cena a možnost řízení libovolným mikropočítačem s UART sběrnici vedlo k rozšíření používání této sběrnice.[4]

#### 1.3.1 Princip komunikace sběrnice

Jedná se o sběrnici typu single master, multiple slave, to znamená že master řídí a kontroluje provoz na sběrnici. Zároveň může plnit funkci gateway mezi LIN sběrnici a sběrnici vozidla. Zajišťuje komunikaci s ostatními podřízenými uzly, kterých může být až 16, dohromady i s master uzlem tedy 17. Maximální přenosová rychlost této jednovodičové sběrnice dosahuje až 20 kbit/s při čemž standardizované rychlosti přenosu jsou 2400 bit/s pro pomalý přenos, 9600 bit/s pro středně rychlý přenos a 19200 bit/s pro rychlý přenos. Synchronizaci provádí vždy master na počátku každého přenosu.[5]

Master uzel vždy obsahuje master a slave task, slave uzly pak obsahují pouze slave task. Master task vybírá podle plánovací tabulky, která určuje pořadí datových rámců a prodlevy mezi nimi, hlavičky těchto zpráv a ty se pak posílají po sběrnici. Podřízené slave jednotky podle příchozího identifikátoru datového rámce vysílají nebo přijímají data.



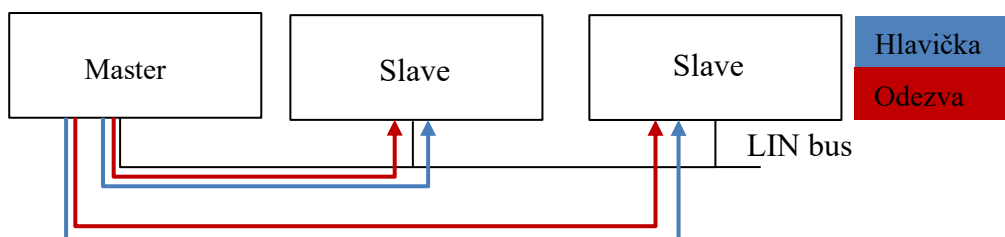
Obrázek 1.4: Příklad komunikační struktury master a slave uzlů na sběrnici

Výčet činností pro master i slave uzly lze rozdělit do následujících bodů:

Master definuje přenosovou rychlost, vysílá synchronizační impuls, synchronizační bajt a identifikátor, monitoruje a potvrzuje data pomocí checksumu, přepíná a probouzí podřízené jednotky do režimu sleep a normálního provozního režimu, reaguje na signál probuzení od podřízených jednotek.

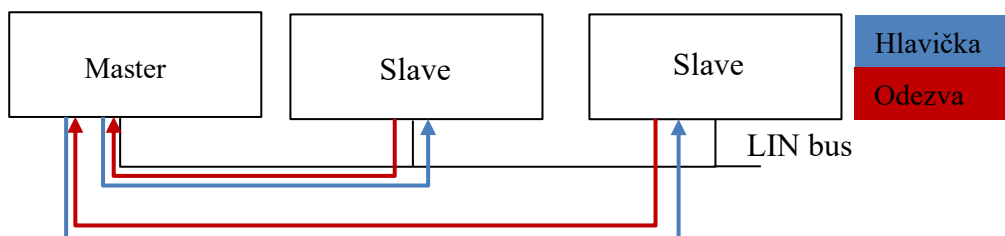
Slave čeká na synchronizační impuls, synchronizuje se podle synchronizačního bajtu, reaguje na identifikátor, a reaguje následujícími způsoby. Odesílá data, přijímá data, nereaguje, kontroluje nebo posílá checksum.

Komunikace mezi jednotlivými uzly může probíhat několika způsoby, viz Obrázek 1.5: je zde znázorněna komunikace master -> slave. Tento typ nastavá například při nastavování motorku do určité polohy nebo nastavování jiných akčních členů. Master vysílá dva rámce, a to příkazový rámec a případně hlavičku pro odezvu slave uzlu.



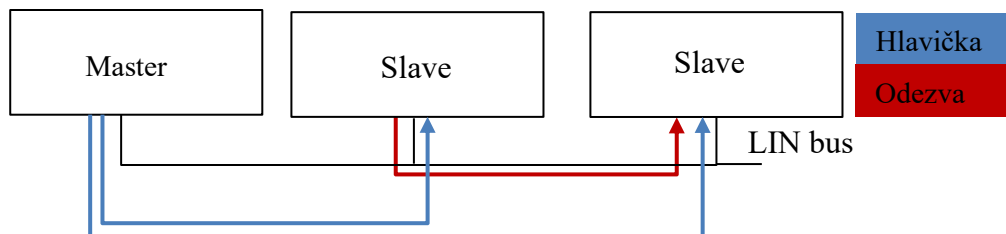
Obrázek 1.5: *Znázornění komunikace master, slave*

Obrázek 1.6: znázorňuje komunikaci slave -> master. Zde master posílá hlavičku stejně jako v předchozím případě, už však nevysílá žádné data, naopak je přijímá od slave uzlů. Tento stav nastává na sběrnici v případě, kdy je třeba zjistit data aktuálně uložené ve slave uzlu. Data pak může master zpracovat a podle nich provést dané zásahy. Stav nastává například, když se řídicí jednotka světlometu ptá na aktuální nastavenou polohu motorku.



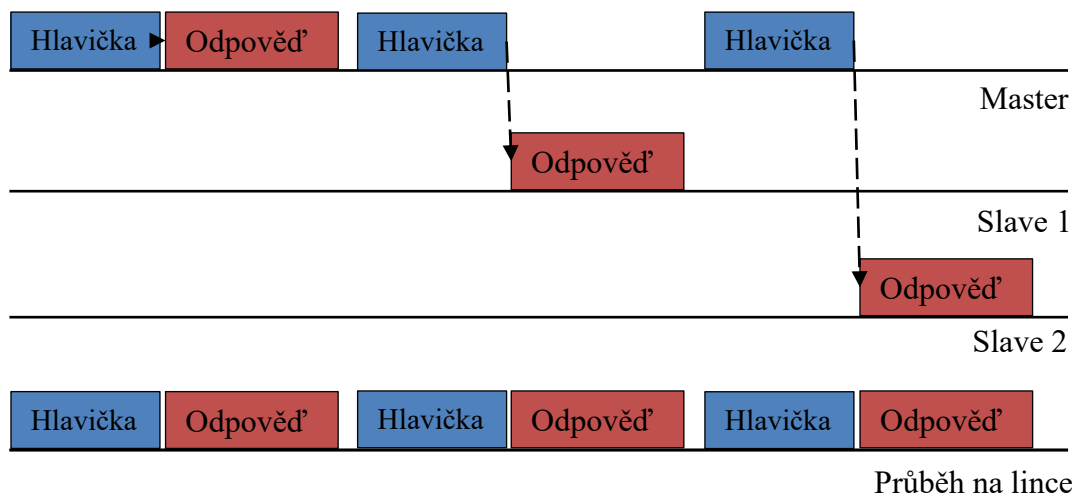
Obrázek 1.6: *Znázornění komunikace slave, master*

Posledním případem komunikace je slave -> slave. Master vyšle hlavičku stejně jako v předchozích dvou případech. Na tento příkaz odpoví slave jednotka datovým rámcem, který master však nepotřebuje a tuto odpověď použije jiný slave.



Obrázek 1.7: Znáznornění komunikace slave, slave

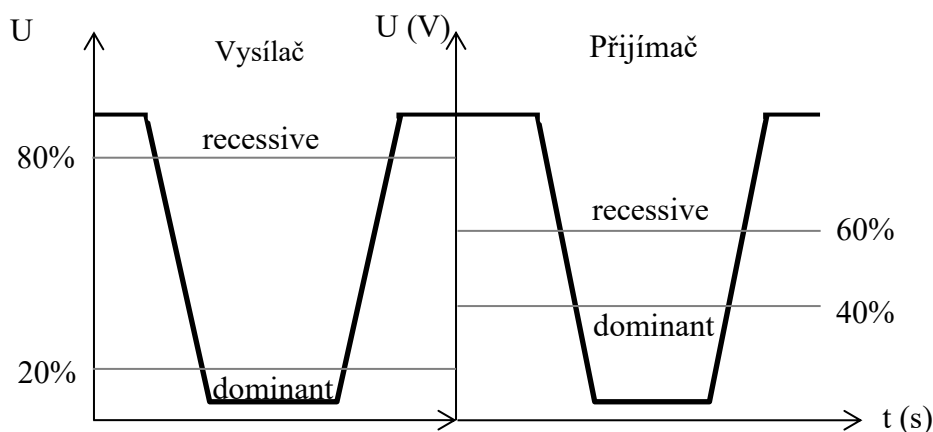
Obrázek 1.8: zobrazuje příklad komunikace master jednotky a dvou slave jednotek. Výsledný průběh na sběrnici je dán posledním řádkem. Každá komunikace je zahájena master jednotkou, vysláním hlavičky. Zdrojem může být samotný master uzel a to znamená, že po odeslání hlavičky zahájí i vyslání dat pro podřízenou jednotku. V druhém případě jsou data od slave jednotky vyžadována, a proto po přijetí hlavičky začne vysílat odpověď. Příjemci zprávy mohou být všechny slave jednotky nebo může být zpráva určena pouze pro jednu konkrétní jednotku.[6]



Obrázek 1.8: Komunikace na sběrnici mezi Master jednotkou a slave jednotkami

### 1.3.2 Hardwarové řešení sběrnice

Fyzická vrstva byla odvozena od standardu ISO 9141. Ten byl vyvinut pro diagnostické účely pro použití v servisech. Pro použití v případě jedoucích automobilů, musely být navrženy určité změny. Například strmost náběžných a sestupných hran je omezena z důvodu minimalizace vyzařovaného rušení. Rozhodovací úrovně se také změnily pro lepší odolnost vůči rušení. Sběrnice LIN používá jeden vodič pro obousměrnou komunikaci, pomocí funkce logického součinu prostřednictvím spínačů a rezistorů zapojených na sběrnici v každém připojeném zařízení. Jsou definovány dvě hodnoty stavů na sběrnici, dominant a recessive. Velikosti a rozsah jednotlivých úrovní jsou vztaženy k palubnímu napětí v automobilu generovanému akumulátorovou baterií 12 V. Hodnoty těchto úrovní jsou popsány viz. Obrázek 1.9: spínače při sepnutí spojují sběrnici se zemí, stačí, aby byl sepnut alespoň jeden z nich, a sběrnice přejde do stavu dominant, což představuje stav logické nuly. Rezistory zapojené mezi napájecí napětí a sběrnici pak na ní udržují, pokud není žádný spínač sepnutý, stav recessive, tedy logickou jedničku.[5]

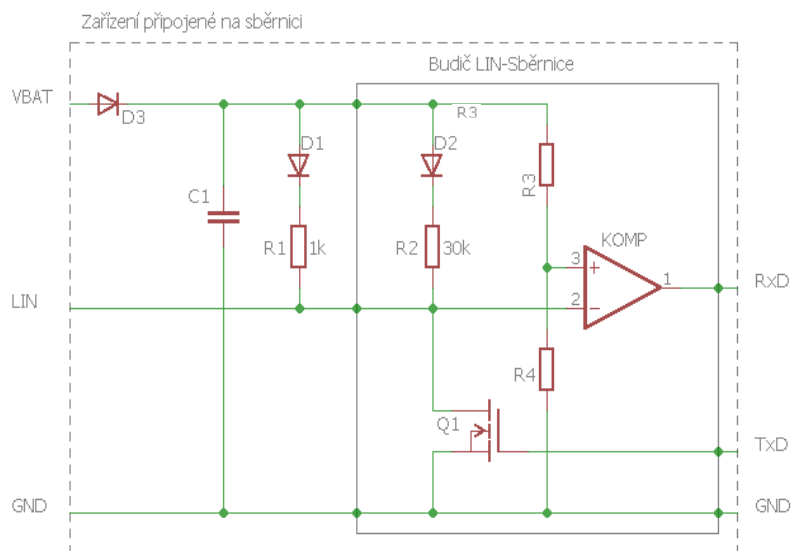


Obrázek 1.9: Znáznornění napěťových úrovní pro vysílač a přijímač

Obrázek 1.9: pro vysílač platí logická 0 do 20 % palubního napětí a logická 1 od 80 % palubního napětí a pro přijímač je definovaná logická 0 do 40 % palubního napětí a logická 1 od 60 %

Obrázek 1.10: znázorňuje zjednodušené schéma zapojení budiče sběrnice. Vodiče VBAT a GND slouží k napájení budiče i ostatních zařízení. Pro případ přerušení napájení zařízení připojeného na sběrnici jsou rezistory definující stav recessive zapojeny v sérii s ochranou diodou. Ta zabrání nedefinovanému napájení jednotky po vodiči LIN. Obecně jsou budiče sběrnice LIN konstruovány tak, aby byly odolné proti poruchovým stavům, které se mohou vyskytnout, jako je zkrat, nebo přepětí. Velikosti těchto rezistorů mají jmenovitou hodnotu 30 k $\Omega$ . Maximální počet zařízení připojených na sběrnici je teoreticky omezen jen počtem volných identifikátorů. Ve skutečnosti je však nutné brát zřetel na elektrické požadavky, které počet striktně omezují na 16 plus master uzel. Aby se s počtem připojených budičů razantně neměnila velikost výsledného odporu připojující sběrnici na napájecí napětí, je definováno, že u zařízení typu master, které je

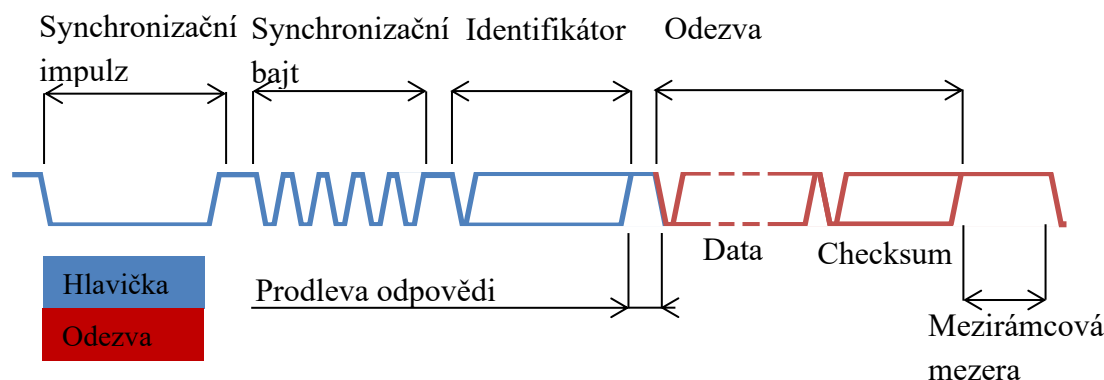
na sběrnici vždy jen jedno, je kromě interního rezistoru, zapojen navíc externí rezistor o hodnotě 1 k $\Omega$ . Z toho též vyplývá, že budiče sběrnice (integrované obvody) jsou pro master a slave stejné a liší se právě jen externími součástkami. Pro zvýšení odolnosti proti elektromagnetickému rušení se paralelně k vývodu LIN budiče připojují kondenzátory.[5]



Obrázek 1.10: *Obecné elektrické schéma budiče sběrnice[6]*

### 1.3.3 Komunikační protokol

Veškerou komunikaci po sběrnici řídí a synchronizuje uzel master. To znamená, že vybírá, kdy začne probíhat komunikace a s jakou přenosovou rychlostí se bude komunikovat, to se zajišťuje tak zvanou hlavičkou zprávy, která vždy a pouze může přijít z tohoto uzlu. Hlavička začíná synchronizačním impulzem, který je dlouhý minimálně 13 bitů a po něm následuje synchronizační bajt. V tomto bajtu se se synchronizují podřízené slave uzly na správnou přenosovou rychlost. Maximální přenosová rychlost 20 kbit/s byla určena s ohledem na elektromagnetickou kompatibilitu. Její hodnota nesmí být překročena pro správnou funkci sběrnice a ostatní elektroniky v okolí vedení sběrnice. Identifikátor slouží k adresaci zprávy, je zde uložena informace o délce datové části a o tom kdo je příjemce zprávy. Identifikátor lze rozlišit z kombinace 6 bitů, které se dělí do různých kategorií podle svého významu a účelu použití. Mezi hlavní skupiny řadíme identifikátory pro obecný přenos dat, pro přenos diagnostických dat, anebo adresy rezervované uživatelem pro definovaný rámec. V odpovědi jsou pak uložena data, nastavená mastrem, nebo data které master potřebuje od slave uzlu zjistit. Na konci každé zprávy je kontrolní součet, který signalizuje, zda zpráva byla nebo nebyla přijata. Pokud dojde k přerušení komunikace více, než na 1,3 sekundy sběrnice se automaticky přepne do stavu sleep, kdy se ji výrazně sníží spotřeba, z tohoto stavu se do normálního režimu vrátí speciálním příkazem.[5]



Obrázek 1.11: Příklad rámce zprávy komunikačního protokolu

## 1.4 Krokové motory

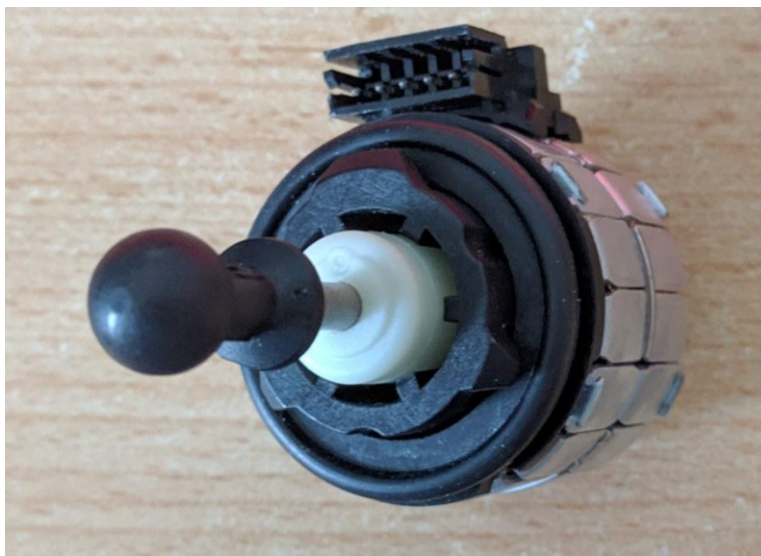
Obecně krokové motory můžeme zařadit do skupiny synchronních motorů, které řídíme buď bipolárně, nebo unipolárně. Způsob řízení se odvíjí od jejich provedení. Celkem máme tři skupiny, a to motory s pasivním rotorem nazývané také reluktanční, tyto motory mají vyjádřené póly na statoru i rotoru a hlavní podmínkou pro jejich funkci je rozdílný počet pólů na statoru a na rotoru, většinou se používají konstrukce 4:6 nebo 6:8, tyto čísla značí počet pólů na rotoru a statoru. Dále mohou být krokové motory s aktivním rotorem, u těchto typů je rotor tvořen permanentním magnetem, tyto motory mají rovněž rozdílný počet pólů na statoru a rotoru. Poslední skupinu tvoří krokové motory hybridní, které slučují principy obou zmiňovaných typů.

Řízení krokových motorů může být tedy unipolární a bipolární. Unipolární řízení je řízení s pouze kladnou nebo pouze zápornou polaritou proudu a toto řízení se používá u motorů s rotorem pasivním. Bipolární řízení se používá u motorů s rotorem aktivním, jelikož rotor tvořený permanentním magnetem má na pevně definovaný severní a jižní pól, je potřeba řešit polaritu proudu. Tyto řízení pak dále dělíme na čtyř – taktní, osmi – taktní, nebo N – taktní tím už odlišujeme pouze velikost a přesnost kroku, řízení a také náročnost řídicího algoritmu.

Krokové motory použité pro aplikaci nastavení světlometů jsou vybaveny speciálním řadičem komunikujícím po sběrnici LIN-bus s řídicí jednotkou, ta se stará o adaptivní svícení světlometu, brání oslnění protijedoucích vozidel a pomáhá také přisvícení do zatáček.

U každého světlometu je potřeba zajistit nastavování úrovně osvětlení, tu zajišťuje levelingový krokový motor JE MA-ELD482. Tento motor slouží k automatickému nastavení lampy v horizontálním směru pomocí senzorů v přední a zadní nápravě vozidla. V případě poruchy motoru je vyslaná chybová hláška do řídicí jednotky a motor se nastaví do nouzové polohy tak, aby projektor neoslňoval okolí.





Obrázek 1.12: *Levelingový krokový motor pro nastavení vertikálního sklonu*

Druhý motor, starající se o pohyb v horizontálním směru při dosvětlování v zatáčkách, se nazývá Sirius EEB721. Tento bendingový motor funguje na stejném principu jako konstrukčně větší levelingový motor, liší se však tvarem a použitím, také obsahuje svou řídicí jednotku, díky které pak komunikuje s hlavní řídicí jednotkou po LINu, přímo ve svém pouzdře. Po startu se nastaví do referenční polohy, která je uložena v paměti řídicí jednotky, následně se nastaví do požadované polohy, která je v určitém intervalu neustále kontrolována. V případě poruchy stejně jako předchozí motor vygeneruje chybovou hlášku a nastaví se do nouzové polohy. Motorky pro svou komunikaci po sběrnici používají protokol v1.3 a vždy čekají na požadavek master uzlu až jim dovolí komunikovat.



Obrázek 1.13: *Bendingový krokový motor pro nastavení horizontálního přisvětlení*

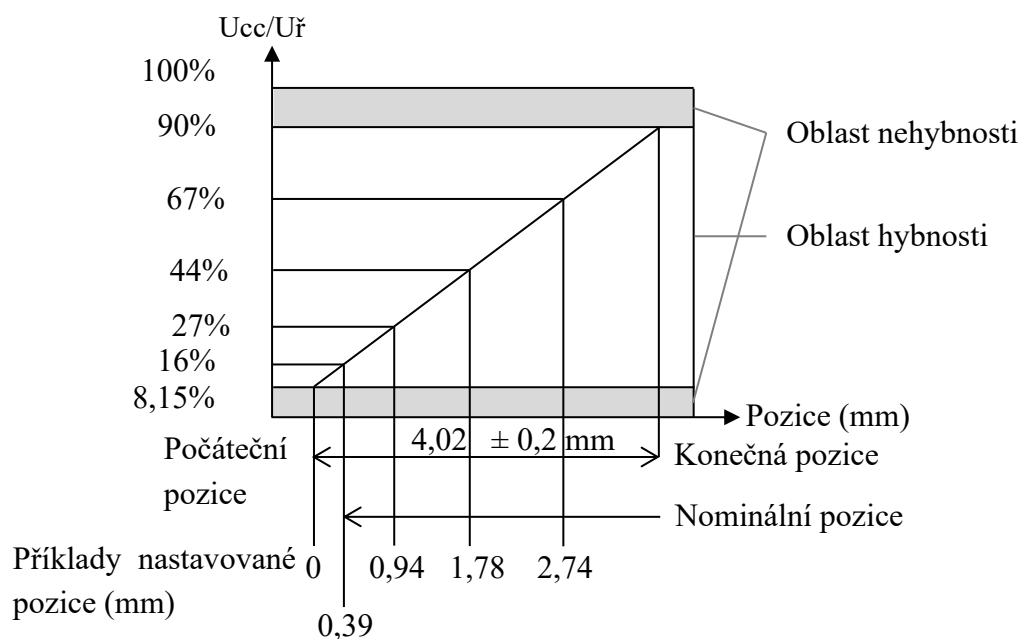
## 1.5 Stejnosměrné motory

Vlastní skupinu tvoří stejnosměrné motory. Svůj název mají díky stejnosměrnému proudu, kterým jsou řízené. Jedná se o nejméně náročné motory na řízení. Otáčky motoru jsou přímo úměrné proudu přiloženému na svorky motoru. Důležitým prvkem stejnosměrného motoru je komutátor, který zajišťuje změnu smyslu proudu v rotorovém vinutí, díky kterému se rotor trvale otáčí jedním směrem. Stejnosměrný motor může být cize buzený, nebo je jeho stator tvořen permanentními magnety. Motory s cizím buzením mají velkou výhodu v oddělené řízení nabuzení budícího vinutí a vinutí kotvy, to znamená, že při tomto typu řízení můžeme řídit moment stroje nezávisle na velikosti jeho nabuzení. Pro servopohony se však nejčastěji používají právě motory s permanentními magnety. Tyto motory se vyznačují velkou proudovou přetížitelností, a to až desetinásobkem nominální hodnoty proudu, který motor odebírá v ustáleném stavu. Používají se v malých a středních výkonech. Motory s permanentními magnety mají příznivý poměr výkonu ke hmotnosti a jsou velmi cenově dostupné. V porovnání s krokovými motory je kvůli komutátoru stejnosměrný motor zdrojem elektromagnetického rušení a také náročnější na řízení a regulaci polohy, rovněž také není bezúdržbový, jelikož zde vzniká jiskření a je třeba dbát na stav uhlíkových kartáčů.



Obrázek 1.14: *Levelingový stejnosměrný motor pro nastavení vertikálního sklonu*

Používaný motor právě jako levnější a méně přesná alternativa krokového motorku je stejnosměrný servopohon od firmy Renault. Tento servopohon má na svém vstupním konektoru celkem tři piny. Dva jsou pro klasické napájení, které se v automobilové elektronice pohybuje okolo 12 V, a třetí řídící pin slouží k nastavení vysunutí polohy hřídele. Motorek má svou vlastní funkční charakteristiku, kdy právě při určité hladině napětí přiložené k řídícímu pinu se vysune do dané polohy. Tato hladina napětí je vztažena k hladině na napájecích pinech a přepočítává se na procenta.



Obrázek 1.15: Funkční charakteristika vysunutí hřídele motorku – poměr  $U_{cc}/U_{ř}$

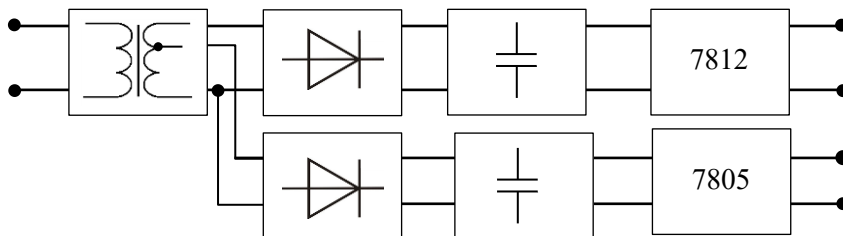
Obrázek 1.15: zobrazuje celkovou dráhu hřídele stejnosměrného servopohonu (osa x), která činí 4,02 mm s tolerancí 0,2 mm a poměr řídicího napětí k napětí napájecímu (osa y). Nominální polohy, při které je motorek montován do světlometu, se dosáhne při 16 % hladiny napájecího napětí. Na obrázku je také vidět oblast nehybnosti nazývaná také jako zakázaná oblast, při hladinách řídicího napětí do 8,15 % a od 90 % napájecího napětí nedochází k mechanickým změnám hřídele a nedoporučuje se přivádět takovéto hladiny napětí na řídicí pin motorku. Na obrázku jsou vyznačeny příklady nastavované polohy pro 27, 44, a 67 % řídicího napětí.

## 2 Návrh testovacího stanoviště pro kontrolu elektrických motorů

Smyslem této práce je realizace testovacího přístroje pro vyřazené krokové a stejnosměrné motory, které jsou při výrobním procesu vymontovány ze světlometu a odstaveny na speciální místo určené pro diagnostiku motorků. Přípravek má pomáhat tyto motorky analyzovat a odhalit, jestli byly vyřazeny oprávněně, někdy se totiž může stát, že za špatný výsledek testu, už zkompleťovaného světlometu nemůže samotný motorek, ale například špatný komponent světlometu jako například řídicí jednotka světlometu, PCB modul ovládající určitou funkci nebo mechanicky poškozené nebo znečištěné sklo světlometu. Z tohoto důvodu byl vznesen požadavek na kompaktní přenosný přístroj, který umožní snadnou diagnostiku těchto motorků na potřebném místě, během několika vteřin.

### 2.1 Návrh napájecího zdroje

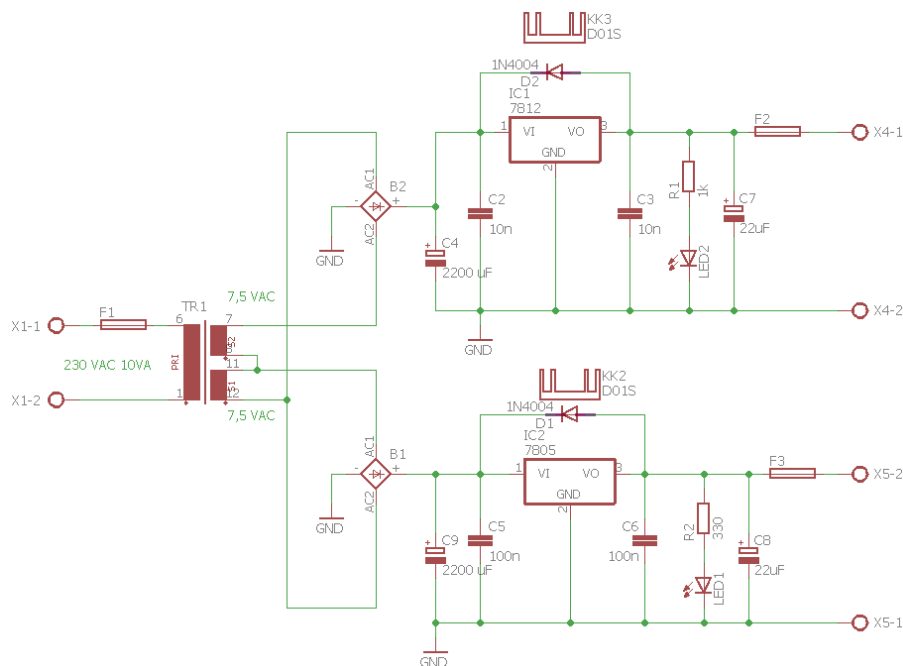
Z důvodu nutnosti mobility tohoto přípravku byl navržen napájecí zdroj, díky kterému není potřeba laboratorní napájecí zdroj a tester motorků tak může být použit kdekoliv, kde se nachází klasická zásuvka. Pro napájení testovacího zařízení z rozvodné sítě byl navržen a použit síťový napájecí zdroj, který slouží k získání stejnosměrných napětí potřebných pro činnost elektronického obvodu, ze kterých je testovací zařízení složeno. Obrázek 1.16: zobrazuje blokové schéma obecného síťového napájecího zdroje. Síťový napájecí zdroj se skládá celkem ze čtyř bloků. Prvním blokem je transformátor, za ním následuje usměrňovač a po té vyhlazovací filtr a jako poslední blok stabilizátor napětí. [7]



Obrázek 1.16: Blokové schéma síťového napájecího zdroje

#### 2.1.1 Elektrické schéma a dimenzace jednotlivých prvků zdroje

Konkrétní návrh napájecího zdroje a jeho elektrické schéma je zobrazeno viz. Obrázek 1.17: Použitý transformátor měl parametry 230 V vstupní napětí se zdánlivým výkonem 10 VA a dvakrát 7,5 V výstupní napětí schopných dodat proud až 666 mA v každé větvi. Pro usměrnění proudu byly použity diodové můstky s filtračním kondenzátorem o velikostech 2200  $\mu F$  pro 12 V větev a pro 5 V větev. Jako stabilizátory byly použity integrované obvody 7812 a 7805 v katalogovém zapojení. Obě větve jsou vybaveny signalizační led diodou a pojistkou jako ochranou proti zkratu.



Obrázek 1.17: Elektrické schéma napájecího zdroje pro tester motorků

Stabilizovaný zdroj stejnosměrného napětí pro napájení testeru se tedy skládá celkem ze dvou větví. První větev je určena pro přímé napájení fyzické vrstvy sběrnice LIN a má poskytnout stejnosměrných 12 V s maximálním proudem až 0,5 A, tato větev bude zajišťovat napájení budiče, který převádí 5 V logiku řídicího mikrokontroléru na hladinu 12 V, která je standardizovaná pro LIN sběrnici.

Druhá 5 V větev je určena pro napájení Arduina, které řídí komunikaci po sběrnici pomocí dříve zmíněného speciálního transciévu TJA1028T, tato větev má poskytnout maximálně 0,5 A, tento proud se označuje jako maximální možný odběr mikrokontroléru při napájení přes USB vstup, ale při zvolené aplikaci se na svorkách neobjeví větší odběr než 100 mA. Hodnota 1 A se uvádí pro napájení arduina z externího zdroje byla zvolena z důvodu maximálního možného proudu, které je Arduino schopno odebírat, aniž by došlo k jeho zničení. [8]

Návrh je rozdělen do několika částí, pro tvorbu elektrických schémat byl použit návrhový CAD software Eagle a pro ověření funkce se provedla simulace v prostředí OrCad, která je dále popsána v následujících kapitolách. Obě větve byly tedy navrženy jako klasický lineární zdroj, připojené k transformátoru o zdánlivém výkonu 10 VA se vstupním vinutím připojitelným až do 230 V, na sekundárním vinutí je 2x7,5 V s maximálním proudem 666 mA. Transformátor s obvodem je na vstupu transformátoru chráněn pojistkou na 63 mA, každou větev pak na výstupu jistí pojistky na proud 800 mA. Jako usměrňovač byl použit diodový můstek B380C1500F(RS207) s propustným napětím  $U_f$  1 V.

Úbytek napětí na 12 V větvi byl potom

$$U_{12RECT} = \sqrt{2} \cdot U_{12vst} - 2 \cdot U_f = \sqrt{2} \cdot 15 - 2 \cdot 1 = 19,21 \text{ V} \quad (2.1)$$

A jeho výkonová ztráta byla

$$P_{12RECT} = (U_{12vst} - U_{12RECT}) \cdot I_{12max} = (20,21 - 19,21) \cdot 0,5 = 1 \text{ W} \quad (2.2)$$

Na 5 V větvi byl úbytek napětí

$$U_{05RECT} = \sqrt{2} \cdot U_{05vst} - 2 \cdot U_f = \sqrt{2} \cdot 7,5 - 2 \cdot 1 = 8,61 \text{ V} \quad (2.3)$$

A jeho výkonová ztráta byla

$$P_{05RECT} = (U_{05vst} - U_{05RECT}) \cdot I_{05max} = (10,61 - 8,61) \cdot 0,5 = 1 \text{ W} \quad (2.4)$$

Pro nízký ztrátový výkon na usměrňovacích můstcích nebylo potřeba navrhnout a vypočítat jejich chlazení.

Výpočet filtračního kondenzátoru pro 12 V větev

$$C_{12} = \frac{I_{12max}}{2 \cdot f(U_{12RECT} - U_{12min})} = \frac{0,5}{2 \cdot 50(19,21 - 14)} = 943,3 \mu F \quad (2.5)$$

Pro tuto větev byl zvolen filtrační kondenzátor o velikosti 2200 uF.

Výpočet filtračního kondenzátoru pro 5 V větev

$$C_{05} = \frac{I_{05max}}{2 \cdot f(U_{05RECT} - U_{05min})} = \frac{0,5}{2 \cdot 50(9,3 - 7)} = 2174 \mu F \quad (2.6)$$

Pro tuto větev byl zvolen filtrační kondenzátor o velikosti 2200 uF. [9]

Jako stabilizátory napětí byly pro 12 V větev použit integrovaný stabilizátor napětí STMicroelectronics 7812CV-DG STM a pro 5 V větev stabilizátor STMicroelectronics 7805CV-DG-STM. Pro jejich chlazení byly vypočteny tyto chladiče.

Stabilizátor 7812

$$P_{12} = (U_{12RECT} - U_{12výst}) \cdot I_{12max} = (19,21 - 12) \cdot 0,5 = 3,606 \text{ W} \quad (2.7)$$

Z datasheetu byla zjištěna hodnota tepelného odporu  $R_{12thj-c}$  pro pouzdro TO-220 5 K/W a maximální teplota  $T_j$ , ve které je obvod schopen pracovat je 150 °C

$$T_{12c} = T_j - (R_{thj-c} \cdot P_{12}) = 150 - (5 \cdot 3,606) = 131,98 \text{ K}$$

Tepelný odpor chladiče  $R_{thc-a}$  je potom:

$$R_{12thc-a} = \frac{T_{12c} - T_a}{P_{12}} = \frac{131,98 - 40}{3,606} = 25,5 \text{ K/W} \quad (2.9)$$

Stabilizátor 7805

$$P_{05} = (U_{05RECT} - U_{05výst}) \cdot I_{05max} = (8,606 - 5) \cdot 1 = 3,606 \text{ W} \quad (2.10)$$

Z datasheetu byla zjištěna hodnota tepelného odporu  $R_{thj-c}$  pro pouzdro TO-220  $5 \text{ K/W}$  a maximální teplota  $T_j$ , ve které je obvod schopen pracovat je  $150 \text{ }^{\circ}\text{C}$

$$T_{05c} = T_j - (R_{thj-c} \cdot P) = 150 - (5 \cdot 3,606) = 131,98 \text{ }^{\circ}\text{C} \quad (2.11)$$

Odpor chladiče  $R_{thc-a}$  je potom:

$$R_{05thc-a} = \frac{T_c - T_a}{P} = \frac{131,98 - 40}{3,606} = 25,5 \text{ K/W} \quad (2.12)$$

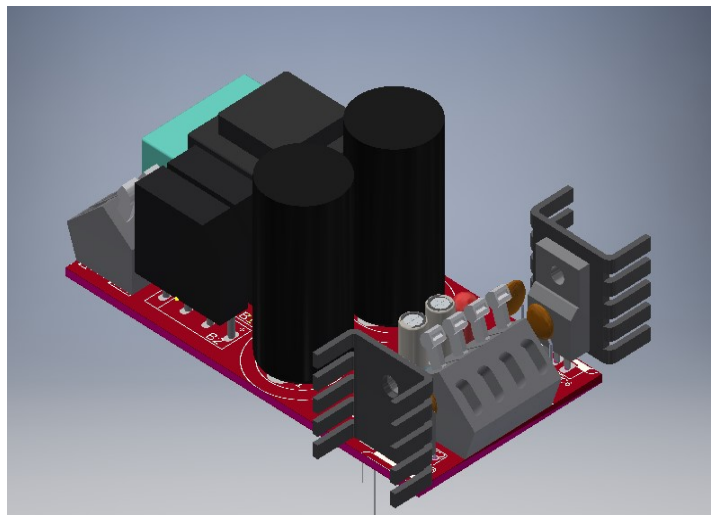
Pro oba stabilizátory byl vybrán chladič D01 s tepelným odporem  $21 \text{ K/W}$  vhodný pro pouzdro IO TO220 [10]



Obrázek 1.18: *Obrázek použitého chladiče [11]*

### 2.1.2 Návrh 3D modelu napájecího zdroje

V prostředí Autodesk Inventor byl navržen 3D model napájecího zdroje skládajícího se z reálných součástek viz. Obrázek 1.19: . Jsou zde modely stabilizátorů 7805 a 7812, vybrané kondenzátory, transformátor, usměrňovací můstky a všechny ostatní použité součástky v reálných rozměrech.

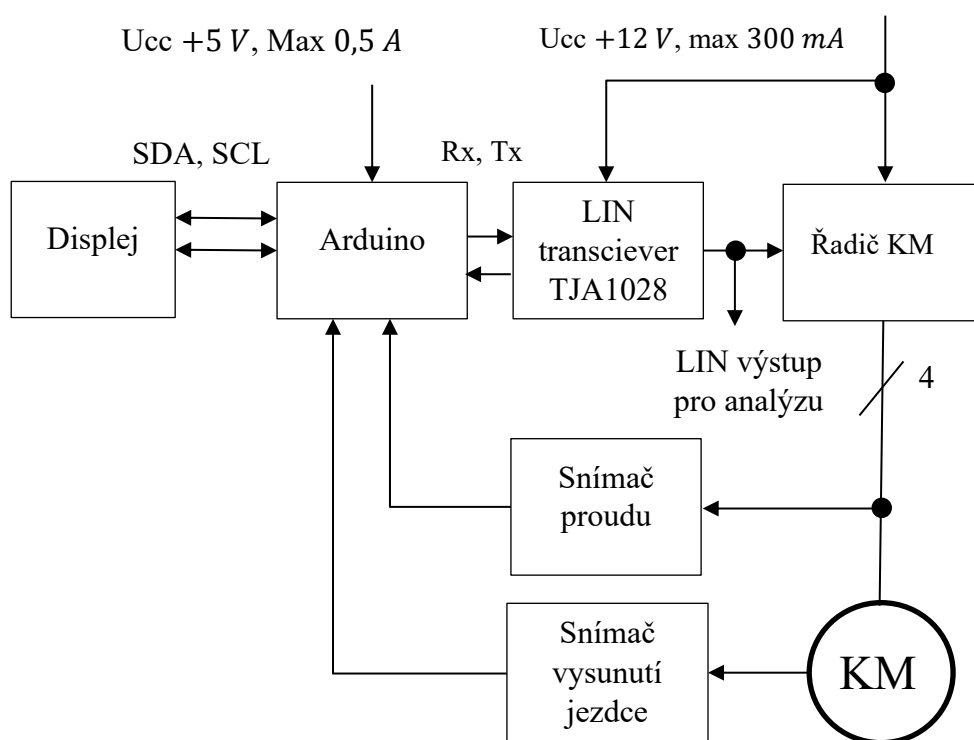


Obrázek 1.19: *3D model napájecího zdroje přípravku*

## 2.2 Návrh testovací části přípravku pro krokový motor

Pro krokový motor byla vyvinuta speciální část testeru, která zvládá komunikovat a ovládat motorek po datové sběrnici LIN. Tester se skládá z několika částí viz. Obrázek 1.20: Jádrem je mikrokontrolér Arduino, které je tvořeno čipem ATmega2560. Tento čip se stará o obsluhu budiče sběrnice, který komunikuje s řídicí elektronikou krokového motorku. Dále arduino přijímá data z měřicího členu, skrze sériové rozhraní.

Část určená pro analýzu krokového motorku se od části pro stejnosměrný motor liší právě zmíněným budičem a řadičem krokového motorku. Části jako displej mikrokontrolér a snímač vysunutí hřídele motoru jsou stejné, jak pro krokový, tak pro stejnosměrný motorek. Vzhledem k odlišné konstrukci stejnosměrného a krokového motorku bylo potřeba upravit konstrukčně přípravek, aby se daly spolehlivě za aretovat a změřit oba motorky.



Obrázek 1.20: Blokové schéma části přípravku pro krokový motor

### 2.2.1 Řídicí část krokového motoru

Jádro celého přípravku tvoří mikrokontrolér platformy Arduino s čipem ATmega2560. Tento mikrokontrolér řídí a měří chování zvoleného elektromotoru podle požadavků, které uživatel předem navolí v UI, tvořeného lcd displejem a tlačítky. Mezi jeho základní funkce patří až celkem 54 digitálních vstupů/výstupů z nich 15 lze ovládat v režimu pulzně šířkové modulace. Čip ATmega2560 byl zvolen hlavně pro větší množství sériových linek, oproti menší, běžněji používané ATmega328, která má k dispozici pouze jednu sériovou komunikaci. V tomto případě



je potřeba minimálně dvou. Jednu pro komunikaci s budičem LIN sběrnice a druhou pro zobrazování měřených hodnot na displeji, který komunikuje s arduinem pomocí sběrnice I2C.

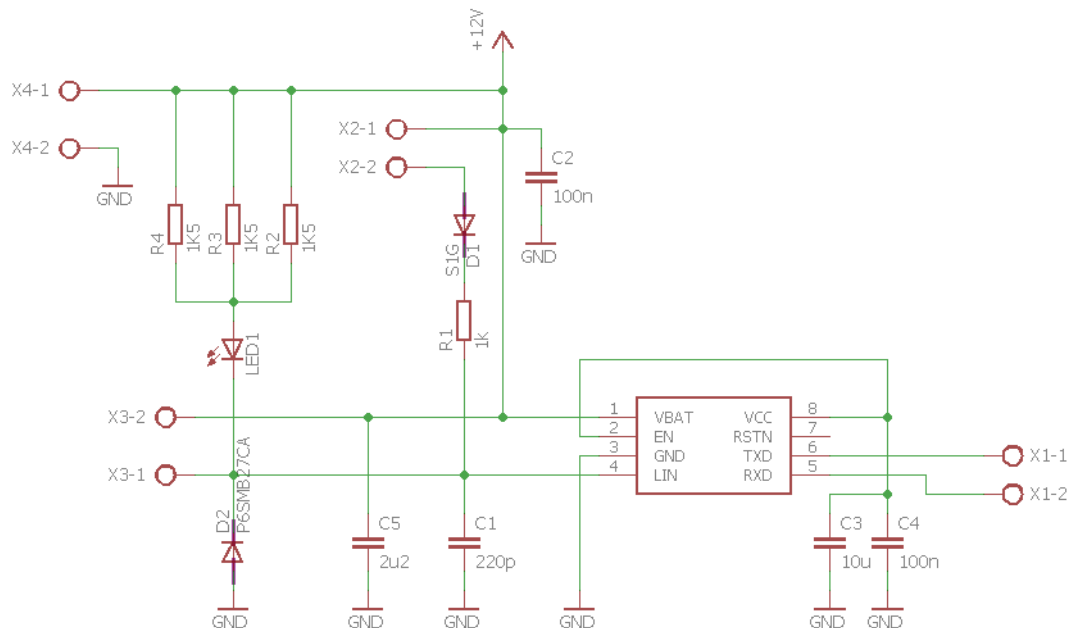


Obrázek 1.21: *Arduino ATmega 2560*[12]

### 2.2.2 Budič sběrnice

Pro komunikaci mezi řídicím čipem a driverem motorku byl použit speciální budič TJA1028T od firmy Philips, který se běžně používá v automotive průmyslu. Jako alternativy se používají také budiče od výrobců Amis, Fujitsu, Infineon nebo Microchip [5]. Tento budič obsahuje napěťový převodník a stará se o snížení napěťové úrovně LIN sběrnice, která je standardizovaná na 12,5 V na hladiny 5 V a 3,3 V. Tento napěťový regulátor je schopný dodat proud o velikosti až 70 mA. TJA1028 usnadňuje vývoj v oblasti LIN systémech a nabízí robustní design a silnou odolnost vůči ESD, je také schopný odolat vysokým napětím na sběrnici a za účelem snížení odběru proudu podporuje TJA1028 takzvaný sleep mode, do kterého se umí po určité době nečinnosti přepnout a vyčkává zde na budicí signál ze sběrnice.

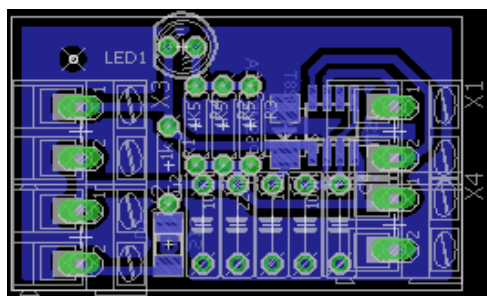
Čip se vyrábí v několika provedeních. Pro tuto aplikaci byl zvolen TJA1028T v pouzdře SO8, díky tomu mohla být deska plošného spoje budiče značně zminimalizovaná. Mezi hlavní funkce a výhody tohoto budiče se řadí kompatibilita s protokolem LIN 2.0, 2.1, 2.2 a zpětná kompatibilita se starší verzí protokolu LIN 1.3. Napěťový regulátor zmíněný v předchozím odstavci na úroveň napětí 5 V a 3,3 V, přesnost těchto napětí při správné provozní teplotě je  $\pm 2\%$ , čip je chráněn proti příliš velké teplotě. Napěťový výstup z regulátoru je vybaven detekcí podpětí s možností restartování výstupu. Budič je chráněn na napájecích pinech proti přechodným jevům, které se mohou vyskytovat v automobilovém průmyslu podle normy ISO 7637. Odběr proudu při sleep módu se pohybuje okolo 12  $\mu A$  a při standby módu okolo 45  $\mu A$ , čip je také kompatibilní s K-line vedením. [13]



Obrázek 1.22: Elektrické schéma zapojení budiče TJA1028T

Na schématu lze vidět napájecí svorky (X4-1, X4-2), jejich připojení je signalizováno LED diodou. Dále jsou zde svorky pro určení master uzlu (X2-1, X2-2) které se nastaví spojením těchto svorek. Svorky jsou hlavně důležité, pokud je na sběrnici více zařízení, které by mohly řídit chod sběrnice. Dále jsou zde svorky, které se připojují na řídicí čip s UART sběrnici (X1-1, X1-2) a větev po které jsou posílána data podle LIN protokolu (X3-1). Budič je chráněn proti přepětí transilem a na svém vstupu a výstupu opatřen filtračními kondenzátory.

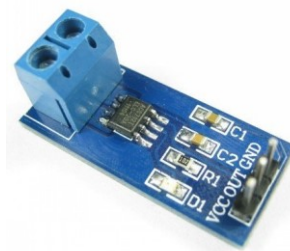
Budič TJA1028T používaný pro komunikaci s krokovými motory tvořící fyzickou vrstvu komunikace sběrnice LIN musí splňovat podmínky komunikačního protokolu vycházející z normy AMIS-30621 LIN Microstepping Motordriver. Pro správnou činnost motoru je cyklicky vykonávána určitá sekvence, která musí zajistit nepřetržitou komunikaci s motorkem. Pro tento budič byla vyrobena vlastní deska plošného spoje viz. Obrázek 1.23:



Obrázek 1.23: Rozložení součástek dps budiče TJA1028

### 2.2.3 Snímač proudu

Pro měření proudu odebíraných oběma typy testovaných motorků byl použit 5 A proudový senzor ACS712 fungující na principu Hallova jevu. Jedná se o aktivní snímač, normalizovaný na pěti voltovou logiku standardizovaný pro Arduino. Mezi jeho hlavní výhody patří měření jak stejnosměrného, tak střídavého proudu s šířkou pásma až 80 kHz a lineární převodní charakteristika. Přestože je tento snímač schopen měřit proud až do 5 A jeho citlivost se udává mezi 66 až 185 mV/A, a to je pro rozeznání bezchybného stavu motorku dostačující. Snímač je velmi kompaktní a jeho provozní teplota se udává od -40 do 85 °C. Jeho chyba je 1,5 %, což při 5 A rozsahu je 75 mA. Při běžném provozu krokový motorek odebírá 500 mA a stejnosměrný motorek 400 mA. Pro přesné měření by tento snímač nemohl být použit, jako indikátor zkratového stavu nebo nulového proudu ale dostačuje. [14]



Obrázek 1.24: Snímač proudu ACS712 [15]

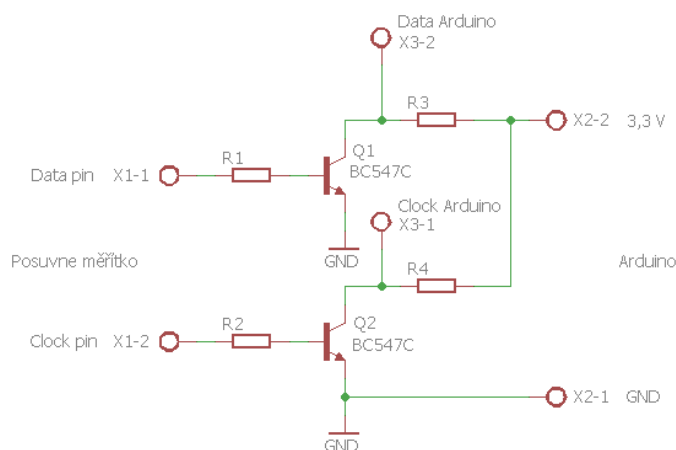
### 2.2.4 Snímač vysunutí jezdce motorku

Pro měření hřídele motorku, kterou je potřeba měřit s přesností na milimetry byl původně zvažován infračervený senzor, který dokáže měřit předměty s potřebnou přesností. Měření předměty tímto senzorem musí ale být prostorově výraznější, než je hlavička hřídele motorku, proto byl jako další možnost snímání hřídele motorku testován lineární potenciometr, který sice přiléhá k hlavičce pomocí speciálního konektoru vytištěného na 3D tiskárně. Při tomto způsobu měření však jeho přesnost nedostačovala a měření tak bylo pouze orientační. Jako nejpresnější a cenově dostupná varianta byla zvoleno měření vzdálenosti pomocí elektronického digitálního posuvného měřítka, které je schopné komunikovat s Arduinem pomocí datového a hodinového pinu.



Obrázek 1.25: Posuvné měřítko s displejem a digitálními vstupy [16]

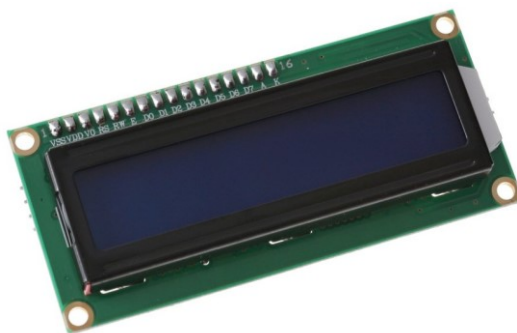
Digitální měřítka jsou běžně napájena z pastilkové 1,5 V baterie. Tato hladina napětí pro Arduino je bohužel nepracovatelná a Arduino nezvládá rozlišit mezi logickou 1 a 0. Z toho důvodu byl použit dva tranzistory ve spínacím režimu řízený datovým a hodinovým signálem z měřítka a čtveřicí 10 k $\Omega$  rezistorů. Toto jednoduché zapojení zajišťuje spínání 3,3 V na výstupu 1,5 V logikou měřítka. Stejného principu je taktéž využito při řízení hladiny napětí 12 V DC motou, kdy na řídicí elektrodu motoru je potřeba přivádět 0 až 12 V.



Obrázek 1.26: Měnič hladiny napětí mezi posuvným měřítkem a Arduinem

### 2.2.5 Zobrazovací jednotka

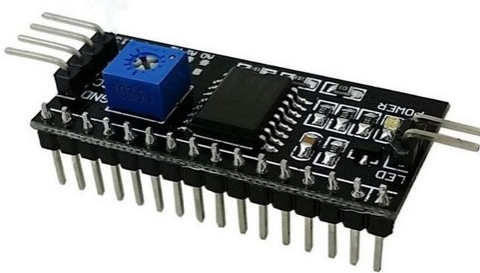
Jako grafický výstup pro zobrazení měřených veličin jako je například průběh vysunutí hřídele je použit LCD displej o rozlišení 16x2 segmenty. Displej samotný má celkem 16 pinů jako je napájecí pin, resetovací pin, zemnicí pin, pin pro nastavení kontrastu displeje a 8 datových pinů.



Obrázek 1.27: Použitý LCD displej s rozlišením 16x2 segmentů [17]

Řízení tohoto displeje takto napřímo je ale zbytečně náročné, proto se pro jeho připojení k Arduino používá obvod PCF8574, který funguje jako osmibitový převodník k I2C sběrnici. Redukuje počet pinů pouze na čtyři, a to na napájecí a zemnicí pin, a pin určený pro data a pin pro hodinový signál. Převodník obsahuje také integrovaný trimr pro nastavení kontrastu displeje.

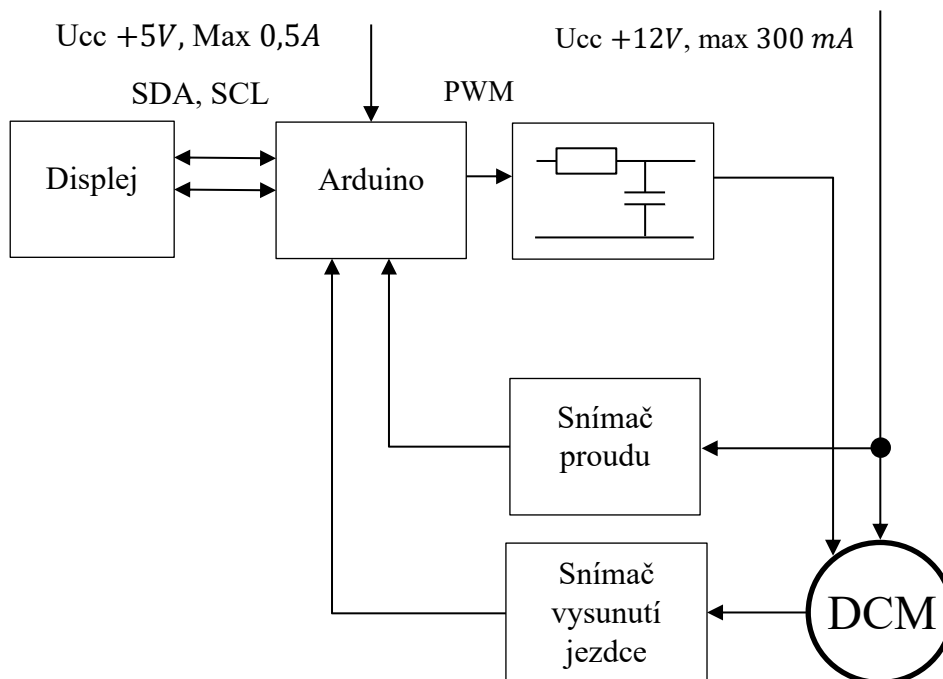
Pro práci s převodníkem je volně dostupná knihovna LiquidCrystal\_I2C, díky které je možné zobrazovat potřebný text pomocí jednoduchých příkazů.



Obrázek 1.28: Převodník k displeji PCF8574 [18]

### 2.3 Návrh testovací části přípravku pro stejnosměrný motor

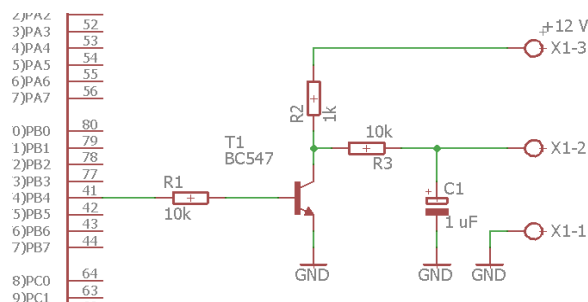
Část přípravku pro stejnosměrný motor se liší od části pro krokový motor právě absencí budiče LIN sběrnice a řadiče krokového motoru. Zobrazovací jednotka a měřící členy jsou stejné pro obě části přípravku. Další velkou odlišností je konstrukční řešení testeru, protože oba motory jsou zcela jinak konstrukčně řešeny, od rozdílného počtu pinů po jejich způsob aretace do přípravku. Obrázek 1.29: poukazuje na odlišnost, že z hlediska struktury vnitřní elektroniky zde oproti části krokového motoru akorát přibyl pasivní filtr prvního řádu, který dělá střední hodnotu z pulzně šířkově modulovaného signálu posílaného na řídicí elektrodu motoru.



Obrázek 1.29: Blokové schéma části přípravku pro stejnosměrný motor

## 2.4 Řídicí část stejnosměrného motoru

Řídicí část přípravku pro stejnosměrný motor je v porovnání s krokovým motorem značně jednodušší, co se týče způsobu řízení motorku. Stejnosměrný motor má předem definované hladiny napájecího a řídicího napětí, podle kterého se má přímo úměrně vysunout hřídel, viz Obrázek 1.15: . Pro řízení tedy dostačuje pulzně šířková modulace výstupu, které je Arduino schopno poskytnout. Bohužel však Arduino pracuje na napětíové úrovni 5 V logiky, a proto je schopno dodat maximálně 5 Voltů na svém výstupu. Tento problém byl vyřešen jako problém příliš nízkého napětí u měřicího členu. Klasický bipolární NPN tranzistor BC547 je řízen PWM výstupem a ovládá se jím externích 12 V, které dostačují pro řízení motorku. Aby se na motorek dostala střední hodnota tohoto napětí, byl použit jednoduchý pasivní filtr dolní propusti prvního řádu viz Obrázek 1.30: .

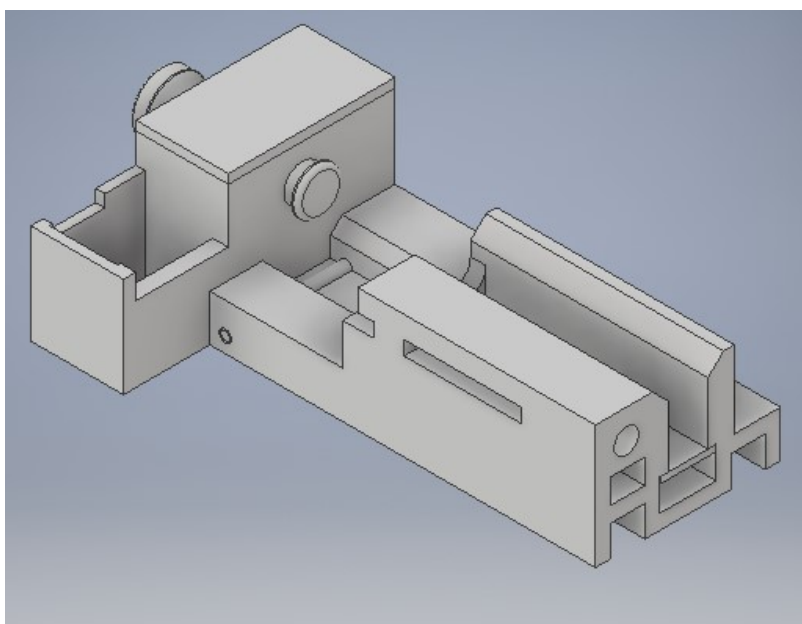


Obrázek 1.30: Měníč hladiny napětí mezi Arduinem a řídicí elektrodou DC motoru

### 3 Realizace testovacího zařízení

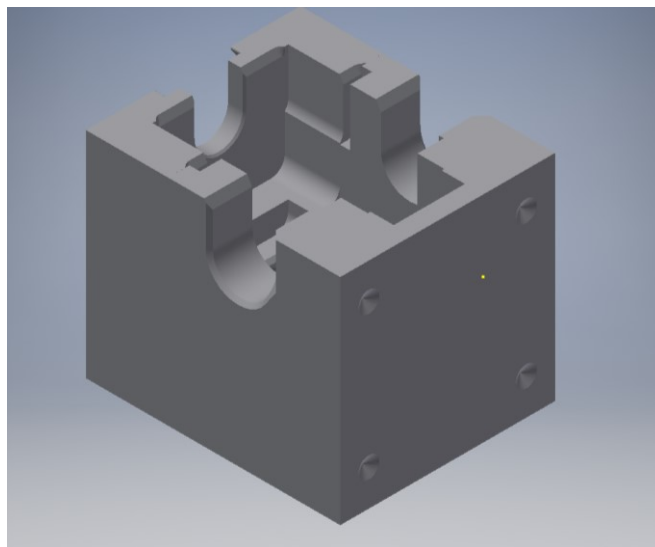
Šasi přípravku byla vytvořena v prostředí Autodesk Inventor a následně vytištěna na 3D tiskárně. Obrázek 1.31: zobrazuje část přípravku navrženou v modelovacím CAD softwaru pro krokový motor, pro motor stejnosměrný je návrh zobrazen viz. Obrázek 1.32: , kompletní sestavení celého testeru je na Obrázek 1.33:

Tato část přípravku slouží pro krokový motor, je zde váleček, který dotlačuje motorek na kontaktovací piny zapuštěné uvnitř konstrukce. Na čelní straně vidíme žlábek, do kterého se uloží posuvné měřítko a úzký otvor obdélníkového tvaru ve kterém se hýbe pohyblivá část měřítka, která je vytlačovaná motorkem jedním směrem a zpětné vrácení pohyblivé části zajišťuje pružinka.



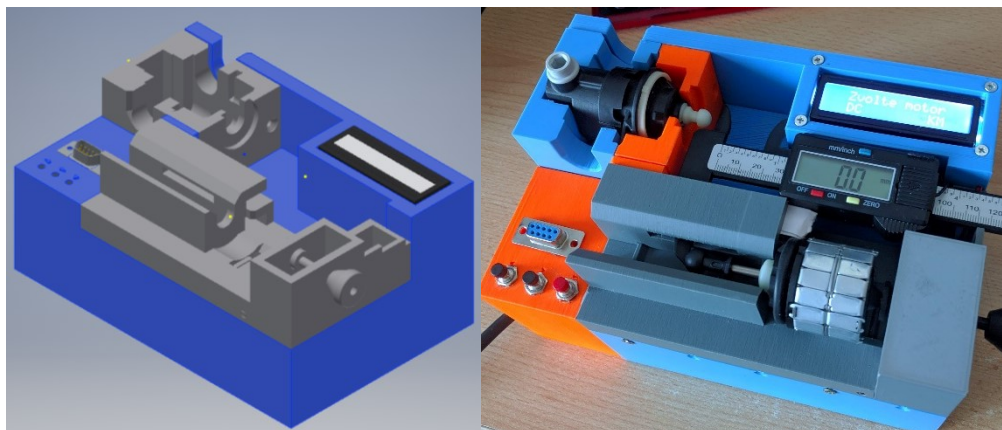
Obrázek 1.31: *Konstrukční část přípravku pro krokový motorek*

Konstrukční část přípravku pro Stejnosměrný motor se liší hlavně tím, že měřený stejnosměrný motor nakontaktování provádí ve vertikální směru. Kontaktovací piny, oproti krokovému motorku jsou kolmé na osu hřídele motorku. U krokového motoru jsou souosé s hřídelí. Vysunutí hřídele pohybuje posuvným měřítkem a aby měřítko na hřídel doléhalo i ve zpětném směru je použita pružina, která tlačí měřítko ve zpětném směru.



Obrázek 1.32: *Konstrukční část přípravku pro stejnosměrný motor*

Obrázek 1.33: ukazuje celkový vzhled přípravku. Při testování se na začátku měření po zapnutí zvolí, který typ motorku bude analyzován pomocí tlačítek a aktuální zobrazená nabídka na displeji uživateli sdělí ve kterém režimu se nachází. Pak už stačí pouze vybrat, zda se chce provést manuální nastavení polohy nebo automatické měření. Po dokončení zvolené sekvence se na displeji objeví výsledek měření, jestli bylo chování motorku správné v případě autotestu. Při manuálním režimu se pouze zobrazuje dráha vysunutí hřídele a tento režim je třeba ukončit ručně.

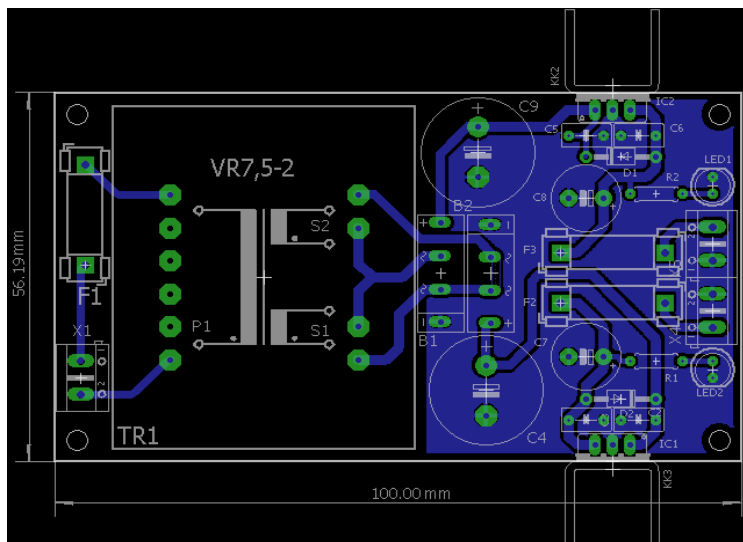


Obrázek 1.33: *3D model a finální verze testovacího přípravku*



### 3.1 Napájecí zdroj testeru elektrických motorků

Při výrobě byla deska plošného spoje ještě mírně upravena oproti návrhu jejího 3D modelu, a proto její výsledná podoba lze vidět viz. Obrázek 1.34: od 3D modelu liší použitím signalizačních diod pro jednotlivé napěťové větve, přidáním děr pro uchycení desky do krabičky testeru a největší změnou byl zde zvolený transformátor, který rozměrově neodpovídal prvotnímu návrhu, a proto bylo desku potřeba celkově zvětšit.



Obrázek 1.34: Rozložení součástek dps napájecího zdroje v programu EAGLE

Pro srovnání je zde přiložena reálná fotografie vyrobeného zdroje viz. Obrázek 1.35: Rozměry zdroje musely být zohledněny při návrhu 3D modelu kompletního testovacího boxu. Protože je celý box z nevodivého materiálu, mohla být použita typická dvojlinka jako přívodní vodič pro napájení zdroje.

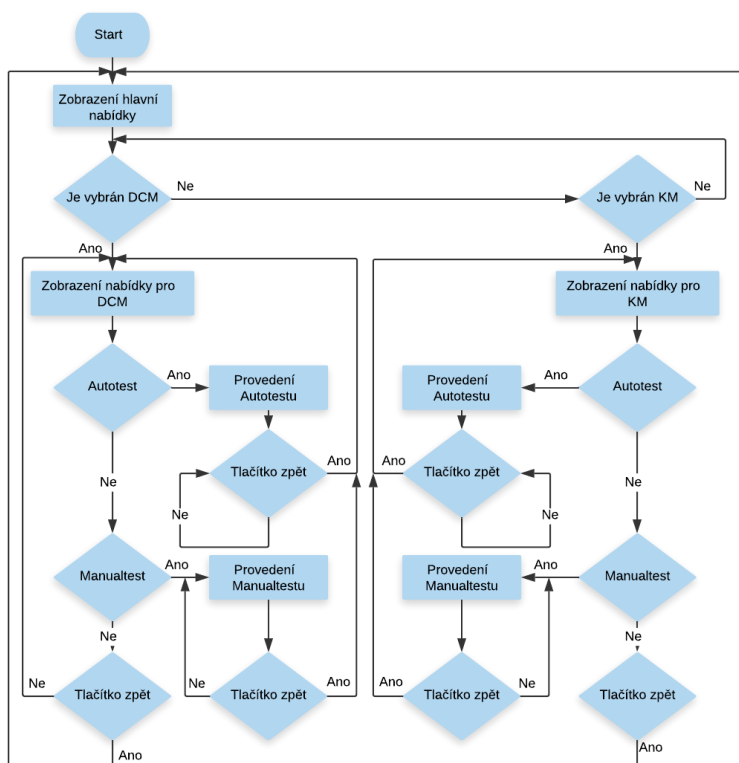


Obrázek 1.35: Vyhotovený napájecí zdroj přípravku

### 3.2 Řídicí software

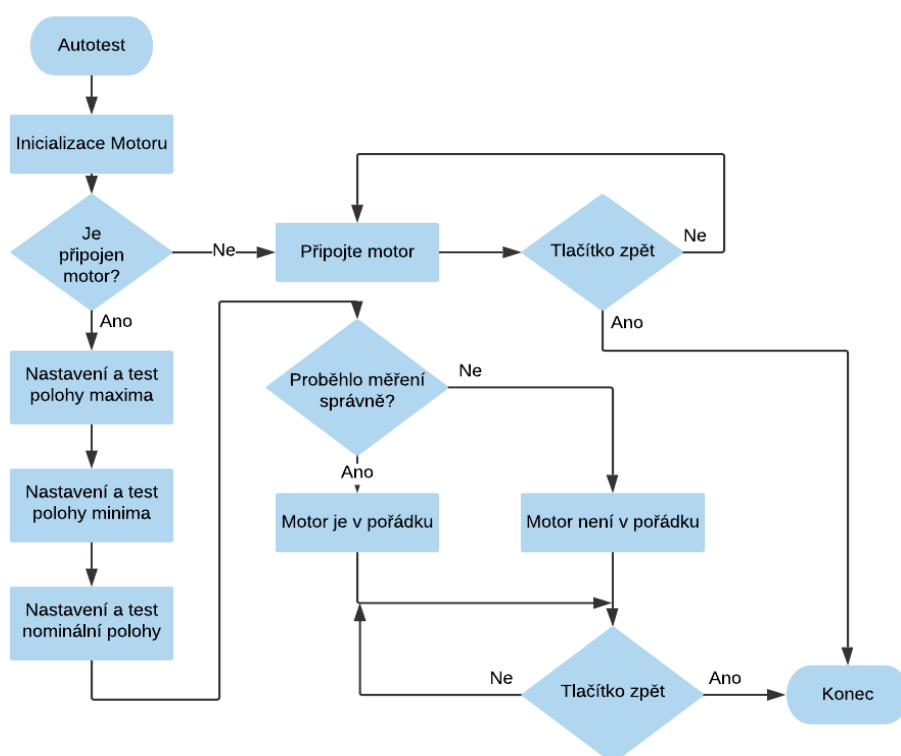
Celý výrobek je řízen platformou Arduino s čipem ATmega2560. Program je napsán v jazyce Wiring, který byl vytvořen speciálně pro tuto platformu. Prostředí, ve kterém byl program napsán, se nazývá Arduino IDE, a je rozdělen do dvou hlavních částí. První část, ve které se nachází funkce setup, se provádí pouze jednou na začátku startu programu po připojení Arduina k napájení. Tato funkce se používá hlavně pro nastavení jednotlivých pinů a definují se zde například počáteční hodnoty proměnných. Druhou částí je funkce loop, která je volána opakovaně, vždy když dojde k jejímu provedení. Potřebný program pro správnou funkci testeru byl však rozdělen do více funkcí a v části loop dochází pouze k jejich opětovnému volání jsou-li splněny dané podmínky. Rozdělení do jednotlivých částí zpřehledňuje program pro možné budoucí úpravy a je celkově čitelnější.

Veškerou nabídku zobrazenou na displeji je možné projít pomocí tří tlačítek, které dávají na výběr z levé, prostřední a pravé části displeje, při čemž v pravé části displeje je vždy umístěna možnost vrátit se zpět, pokud se nenacházíme na domovské obrazovce. Po zapnutí přípravku se na displeji objeví možnost testovaného motoru. Po zvolení typu motoru, který se má testovat, se dále zobrazí nabídka, která je svými následujícími možnostmi totožná pro oba typy motoru a liší se pouze řídicím algoritmem. V této nabídce nalezneme možnost auto testu a manual testu, které jsou popsány dále. Obě varianty je možno opustit a vrátit se na předchozí nabídku pomocí tlačítka zpět.



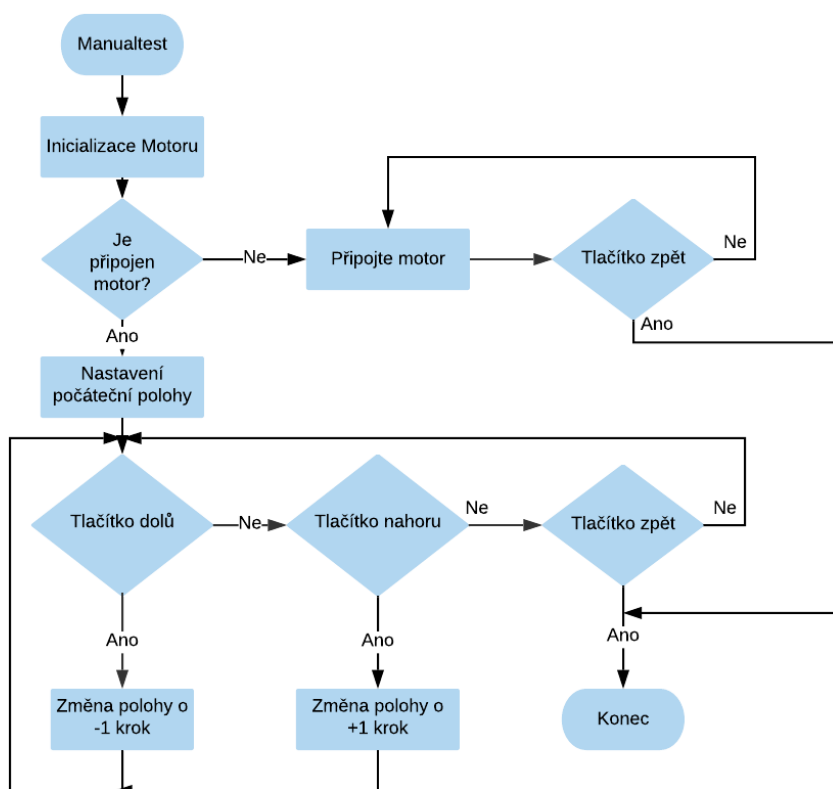
Obrázek 1.36: Průběh programu

Při zvolení možnosti autotest se vykoná určitá sekvence příkazů, která má za úkol proměřit motorek a zkontrolovat jeho stav, zda je vyhovující nebo ne. Na začátku tohoto testu se provede inicializace motoru a pokud jde o krokový motor, nastaví se veškeré parametry potřebné pro jeho správnou funkci (tyto parametry jsou popsány v kapitole 3.3.2). Dále se zkontroluje, zda je zvolený motor připojen, pokud ano pokračuje se na další krok. Pokud se vyhodnotí že je motor nepřipojen dojde k upozornění, aby se provedlo jeho připojení. Po připojení motoru je třeba autotest spustit znovu. Bez připojeného motoru není možné testovací sekvenci provést. Po úspěšné kontrole připojení motoru dochází k provedení samotného auto testu, který má nastavit motor do maximální, minimální a nominální polohy a při tom tento průběh měřit posuvným měřítkem. Pokud se shodují všechny nastavené polohy s hodnotami naměřenými měřicím členem, je motorek vyhodnocen jako fungující a může se vrátit zpátky do výrobního procesu. Na konci testovací sekvence se výsledek zobrazí na displeji a je nutno testovací sekvenci opustit tlačítkem zpět.



Obrázek 1.37: Diagram testovací sekvence autotest

Při zvolení možnosti Manual test je začátek testovací sekvence shodný s předešlým popisem Auto testu. Je zde také inicializace motoru na začátku sekvence a následná kontrola jeho připojení. Pak takto připravený motorek je schopen měnit svou polohu podle stisknutých tlačítek na testeru. Opět motorek může ručně dosáhnout všech poloh jako při autotestu, ale pokud by bylo potřeba motorek nastavit do polohy jiné než nominální, ze které vždy opouští sekvenci Auto test, může se toho docílit zde. Testovací sekvenci je opět možno kdykoliv opustit tlačítkem zpět.



Obrázek 1.38: Diagram testovací sekvence Manual test

### 3.3 Princip komunikace s krokovým motorkem po sběrnici LIN

Podle specifikace AMIS-30621 je potřeba na začátku každé komunikace zajistit inicializaci krokového motorku, ta se ve správném případě provede pouze jednou a nastaví se zde statické, a dynamické adresy. Ty určují činnost, kterou má motor vykonat jako například pohyb, nastavení nominální polohy nebo diagnostika motoru. Po správném přiřazení dynamických identifikátorů se musí nastavit parametry pohybu jako jsou například rychlost a směr pomocí funkce SetMotorParam. [19]

Nyní je motorek připraven ke své funkci a je schopen odpovídat master uzlu na jednotlivé požadavky. Je důležité, aby komunikace s motorkem probíhala cyklicky minimálně co 1,3 sekundy, pokud by na sběrnici nebyla po tuto dobu žádná aktivita, řadič motorku uvede motorek do takzvaného režimu sleep. Pokud přejde motorek do tohoto režimu, nebo dojde k odpojení napájení je potřeba motorek pokaždé nainicializovat znovu. Inicializace se provádí pomocí speciální sekvence zpráv zobrazených v tabulce Tabulka 1.1.; kde proběhne inicializace pouze pomocí statických adres a v tabulce Tabulka 1.2.; je i vyobrazeno nastavení dynamických identifikátorů. K zamezení režimu sleep se po inicializaci cyklicky posílá zpráva GetActPos popsána níže v tabulce Tabulka 1.10:.

Tabulka 1.1: *Princip komunikace s motorkem pomocí statických adres*

	ID	Data master	Data slave
Nastavení parametru motorku	0x3C	SetMotorParam	
Požadavek na vyčtení parametru z motorku	0x3C	GetFullStatus	
Vyčtení parametru z motorku	0x3D		SlaveResponse1
Vyčtení parametru z motorku	0x3D		SlaveResponse2
Cyklické zjišťování polohy motorku co 1,3s	0x3C	GetActPos	
Cyklické zjišťování polohy motorku co 1,3s	0x3D		ActPosMot

Tabulka 1.2: *Princip komunikace s motorkem pomocí dynamických adres*

	ID	Data master	Data slave
Přiřazení dynamických adres	0x3C	DynamicID1 - 4	
Nastavení parametru motorku	0x3C	SetMotorParam	
Požadavek na vyčtení parametru z motorku	0x3C	GetFullStatus	
Vyčtení parametru z motorku	0x3D		SlaveResponse1
Vyčtení parametru z motorku	0x3D		SlaveResponse2
Cyklické zjišťování polohy motorku co 1,3s	0x24		ActPosMot

Tabulky Tabulka 1.1: a Tabulka 1.2: se od sebe liší pouze počátečním nastavením dynamických adres v případě, že se s nimi bude dále pracovat. Při analýze krokových motorků je komunikace pomocí dynamických adres použita. Rozdíl mezi statickými a dynamickými adresami jde nejlépe vidět při posílání jednotlivých příkazů jako například při zjišťování aktuální polohy motorku. Nastavování a zjišťování polohy je zobrazeno v tabulce Tabulka 1.3: pomocí statických adres a v tabulce 0 pomocí dynamických adres.

Tabulka 1.3: *Nastavování a zjišťování polohy pomocí statických adres*

	ID	Data master	Data slave
Nastavení polohy	0x3C	SetPos	
Vyčítání polohy požadavek	0x3C	GetActPos	
Vyčítání polohy odpověď	0x3D		ActPosMot

Tabulka 1.4: *Nastavování a zjišťování polohy pomocí dynamických adres*

	ID	Data master	Data slave
Nastavení polohy	0x08	SetPos	
Vyčítání polohy požadavek+ odpověď	0x24		ActPosMot

Přiřazením dynamických adres můžeme nadefinovat až celkem 16 funkcí které budou mít zkrácený zápis. Tyto funkce budou definovány podle identifikátoru a při jejich používání

nebude potřeba volat nejprve takzvaný preparing frame a následně response frame, ale bude stačit poslat pouze jeden rámeček. V dynamické zprávě je zabalena funkce pomocí identifikátoru spolu s adresou slave uzlu, na který má být vyslána tak usnadní řízení motorku a celkovou komunikaci na sběrnici. Pro využívání tohoto způsobu komunikace je potřeba dynamické adresy přiřadit příkazem Dynamic asist ID, který je jako první použit v tabulce Tabulka 1.2.; podrobně rozepsaný je níže v tabulce Tabulka 1.5:.

### 3.3.1 Dynamické adresování

Zpráva pro přiřazení dynamických adres se dá rozdělit do dvou částí. V první části je klasický identifikátor a první 3 data byty. Ty se skládají z takzvaného application command, který se vyskytuje v každém Master request frame, data byte 2 se nazývá Command byte, ten určuje druh funkce a poslední byte 3 nastavuje adresy motoru a také v bitu s nejvyšší hodnotou se volí, zda je zpráva pro všechny slave zařízení na sběrnici nebo pouze pro adresované uzly. Druhá polovina této zprávy, rozdělena celkem do pěti bajtů, se skládá ze čtyř rom pointerů, které definují funkci nastavovaného ID a pak jsou zde rovněž čtyři dynamické ID, které jsou navoleny uživatelem. To znamená, že každý rom pointer svou hodnotou určuje příkaz, který se má vykonávat například GetActPos, GetStatus nebo SetPos a ke každému rom pointeru do páru se přiřazuje dynamická adresa, jejíž hodnotu určuje uživatel a tato hodnota se v pozici identifikátoru posílá jako součást příkazu, který je na sběrnici poslán.

Tabulka 1.5: Rozpis dat příkazu Dynamic asist ID

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
0	0	1	1	1	1	0	0	ID = 0x3C
AppCMD = 0x80								Data1
1	CMD [6:0] = 0x11							Data2
Board	AD6	AD5	AD4	AD3	AD2	AD1	AD0	ADRData3
DynID_1[3:0]				ROMp_1[3:0]				Data4
DynID_2[1:0]		ROMp_2[3:0]				DynID_1[5:4]		Data5
ROMp_3[3:0]				DynID_2[5:2]				Data6
ROMp_4[1:0]		DynID_3[5:0]						Darta7
DynID_4[5:0]						ROMp_4[3:2]		Data8
Checksum								Checksum

### 3.3.2 Nastavení parametrů

Tabulka 1.6: je součástí inicializačního cyklu při zapnutí testovacího režimu krokového motoru. Tento příkaz se skládá opět z osmi bytů jako předchozí tabulka Dynamic asist ID. Má souhlasný identifikátor, protože se jedná o master request frame a první tři byty si jsou rovněž podobné. Ve čtvrtém bytu nalezneme hodnotu proudu motorku a následuje také hodnota maximální a minimální rychlosti, které je motorek schopen dosáhnout. Dále jsou zde uloženy hodnoty zrychlení, které jsou pevně definovány a musí se používat a je zde parametr který určuje

směr otáčení motorku, zda se rotor bude točit po směru nebo proti směru hodinových ručiček. Celkem deset bitů zabírá informace o takzvané Secure position, do této polohy se motorek nastaví vždy pokud přejde motor do režimu sleep, nebo pokud dostane příkaz, aby se do ní cíleně nastavil. V posledním bytu je pak uložena informace o způsobu zrychlení a velikosti používaného kroku.

Tabulka 1.6: Rozpis dat příkazu *SetMotorParam*

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
0	0	1	1	1	1	0	0	ID = 0x3C
AppCMD = 0x80								Data1
1	CMD [6:0] = 0x09							Data2
Board	ADR [6:0]							Data3
Irun [3:0]				Ihold [3:0]				Data4
VMax [3:0]				VMin [3:0]				Data5
SecPos[10:8]			Shaft	Acc [3:0]				Data6
SecPos [7:0]								Darta7
1	1	1	AccShape	StepMode [1:0]	1	1		Data8
Checksum								Checksum

### 3.3.3 Vyčtení parametrů

Po zavolání příkazu SetMotorParam se následně posílá zpráva GetFullStatus, která se skládá z preparing frame a dvou response frame. Preparing frame vyšle master jednotka na motorek a response frame, rovněž vyslána master jednotkou, očekává od slave jednotky odpověď. Response frame je u tohoto příkazu rozdělen do dvou zpráv a zabírá celkem 16 datových bytů. V preparing frame se pošle pouze požadavek na konkrétní adresu a jsou zde vyžádány veškeré informace, které jsou potřebné pro chod motorku, důležité jsou pouze první tři byty, ve kterých je uložena informace o typu požadavku a jeho adrese na kterou je poslán. Data 4 až data 8 jsou nastaveny na hodnotu 0xFF.

Tabulka 1.7: Rozpis dat příkazu *GetFullStatus – preparing frame*

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
0	0	1	1	1	1	0	0	ID = 0x3C
AppCMD = 0x80								Data1
1	CMD [6:0] = 0x01							Data2
1	ADR [6:0]							Data3
Data4 [7:0] FF								Data4
Data5 [7:0] FF								Data5
Data6 [7:0] FF								Data6
Data7 [7:0] FF								Darta7
Data8 [7:0] FF								Data8
Checksum								Checksum

V první části response frame příkazu GetFullStatus dostáváme informaci o adrese, dále jsou zde uloženy hodnoty proudu motorku v klidovém a provozním režimu a velikosti maximální a minimální rychlosti. Tato část zprávy dává o sobě vědět, zda informace nastavované v příkazu SetMotorParam korespondují a jestli jejich nastavení proběhlo správně. Od 4 datového bytu jsou zde dále upřesňující informace o akceleraci, a režimu kroků kterým je motor řízen, pak také směr otáčení. Další bity informují o jakýchkoliv možných chybách, které se mohly při inicializaci vyskytnout jako je ztráta kroku, zda došlo k odpojení napětí nebo jinému defektu elektrického okruhu jako například přerušení kontaktu s cívkou motoru. Jsou zde také uloženy informace o teplotě, jestli nebyla překročena bezpečná teplotní hranice, při které je možno motor provozovat.

Tabulka 1.8: Rozpis dat příkazu GetFullStatus – Response frame 1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
0	0	1	1	1	1	0	1	ID = 0x3D
1	AD [6:0]							Data1
Irun [3:0]				Ihold [3:0]				Data2
VMax [3:0]				VMin [3:0]				Data3
AccShape	StepMode [1:0]		Shaft	Acc [3:0]				Data4
Vddreset	StepLoss	ElDef	UV2	TSD	TW	Tinfo [1:0]		Data5
Motion [2:0]			ESW	OVC1	OVC2	1	CpFail	Data6
1	1	1	1	TimeE	DataE	HeadE	BitE	Data7
0xFF								Data8
Checksum								Checksum

V druhé části response frame dostáváme už jen aktuální adresu jezdce motoru, ve které se nachází, tato informace zabírá celkem 16 bitů a je proto rozdělená vždy do dvou bytů. Data bytu 2 a 3 je informace o aktuální poloze, v bytu 4 a 5 se mohou informace o poloze shodovat s informacemi z bytu 2 a 3 a v bytu 6 a částečně 7 dostaneme informaci o takzvané secure position kde se motor nastaví vždy pokud dojde k přerušení komunikace nebo je do ní účelně poslán. Zbylé bity v 7 bytu spolu s bytem 8 nesou informaci o pozici, kde došlo k zastavení hřídele.

Tabulka 1.9: Rozpis dat příkazu GetFullStatus – Response frame 2

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
0	0	1	1	1	1	0	1	ID = 0x3D
1	AD [6:0]							Data1
ActPos [15:8]								Data2
ActPos [7:0]								Data3
TagPos [15:8]								Data4
TagPos [7:0]								Data5
SecPos [7:0]								Data6
StallPos [11:8]				1	SecPos [10:8]			Darta7
StallPos [7:0]								Data8
Checksum								Checksum



### 3.3.4 Zjištění aktuální polohy pomocí dynamických adres

Příkaz GetActPos s přímým nebo takzvaným dynamickým identifikátorem cyklicky zjišťuje polohu motorku, aby nedošlo k přechodu do sleep režimu. Tato verze dynamického identifikátoru se liší použitím pouze jedné zprávy namísto dvou, kdy není třeba vysílat nejprve preparing frame a následně response frame. Jak bylo popsáno v Tabulka 1.5., identifikátorem se určuje, o jaký příkaz se jedná a v prvním bytu je uvedena informace, kde je příkaz adresován. Hodnota identifikátoru je určena opět v Tabulka 1.5: pomocí dynamického rom pointeru jehož hodnota je předem daná výrobcem. V příkazu GetActPos najdeme tedy dále informaci o aktuální poloze a poslední byte nese informaci o teplotě, poruše elektrického okruhu, ztrátě kroku a velikosti napětí, popřípadě jeho nedostatku.

Tabulka 1.10: Rozpis dat příkazu GetActPos – Direct ID verze

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
X	X	1	0	ID3	ID2	ID1	ID0	ID
ESW	AD [6:0]							Data1
ActPos [15:8]							Data2	
ActPos [7:0]							Data3	
VddReset	StepLos	ElDef	UV2	TSD	TW	Tinfo [1:0]		Data4
Checksum							Checksum	

Princip přiřazení dynamické adresy podle hodnoty v rom pointeru je znázorněn v Tabulka 1.11: Zde lze vidět že hodnota rom pointeru má napevno definovaný druh příkazu. O jaký druh příkazu se jedná je určeno výrobcem a je zobrazen v prvním sloupci. V druhém sloupci je zobrazena opět výrobcem určená hodnota rom pointeru a ve třetím sloupci pak uživatel nastaví hodnotu dynamického identifikátoru, který bude za zvoleným účelem používat. Tabulka svými hodnotami koresponduje s tabulkou, kde se nastavují dynamické identifikátory.

Tabulka 1.11: Legenda k přiřazení dynamických adres

Druh příkazu	ROM pointer	Dynamické ID
SetPos	0100	0x08
GenPurp	0001	0x20
GetActPos	0010	0x24
GetStat	0011	0x04

### 3.4 Vector CANoe

CANoe je univerzální nástroj pro vývoj, testování a analýzu celých automobilových sítí i jednotlivých ECU. Podporuje například sběrnice CAN, LIN, FlexRay, K-line a další. CANoe se využívá zejména vývojovými a testovacími inženýry během celého procesu vývoje od plánování až po spuštění a údržbu všech distribuovaných systémů a jednotlivých ECU. Pro vývoj testeru byla nejdůležitější analýza sběrnice LIN.

Na začátku procesu vývoje se CANoe používá k vytváření simulačních modelů, které simulují chování systému pro ECU. V průběhu dalšího vývoje tyto modely slouží jako základ pro analýzu, testování a integraci automobilových systémů a ECU. To umožňuje včasné odhalení problémů a jejich opravu. Pro vyhodnocení výsledků jsou k dispozici grafická okna i textová analýza.

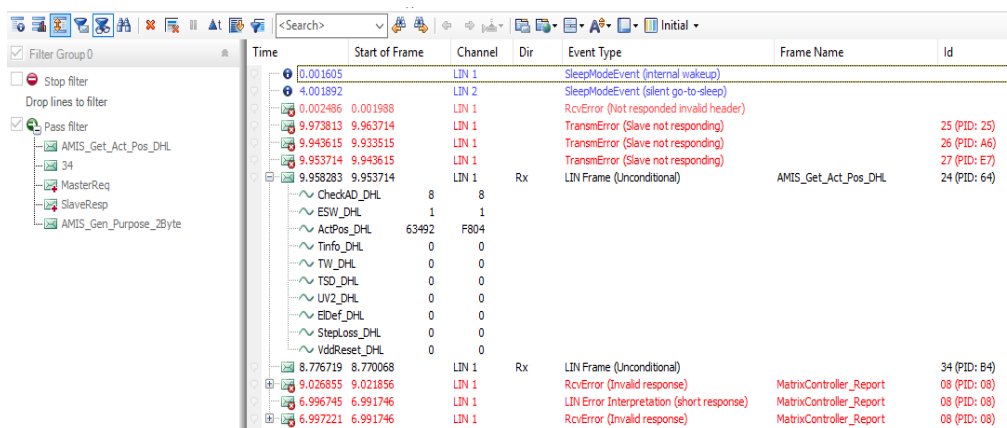
CANoe obsahuje set testovacích funkcí pro snadné a automatizované provádění testů. Používá se k modelování a provádění celých testovacích sekvencí a jejich automatickému generování zprávy o provedeném testu. K dispozici je v CANoe také diagnostická sada funkcí pro diagnostiku komunikace s ECU. [20]

Mezi základní funkce patří práce s databázemi, které popisují konkrétní síť. Jde o soubory typu DBC, FIBEX, LDF, a další. V této práci byl použit soubor LDF, díky kterému bylo možné analyzovat komunikaci probíhající mezi ECU a jednotlivými akčními členy sběrnice LIN. Zprávy probíhající na sběrnici se zobrazují v takzvaném trace window, jsou barevně odlišeny pro lepší orientaci a dají se seřadit či úplně skrýt pomocí zvoleného filtru.

#### 3.4.1 CANoe.LIN

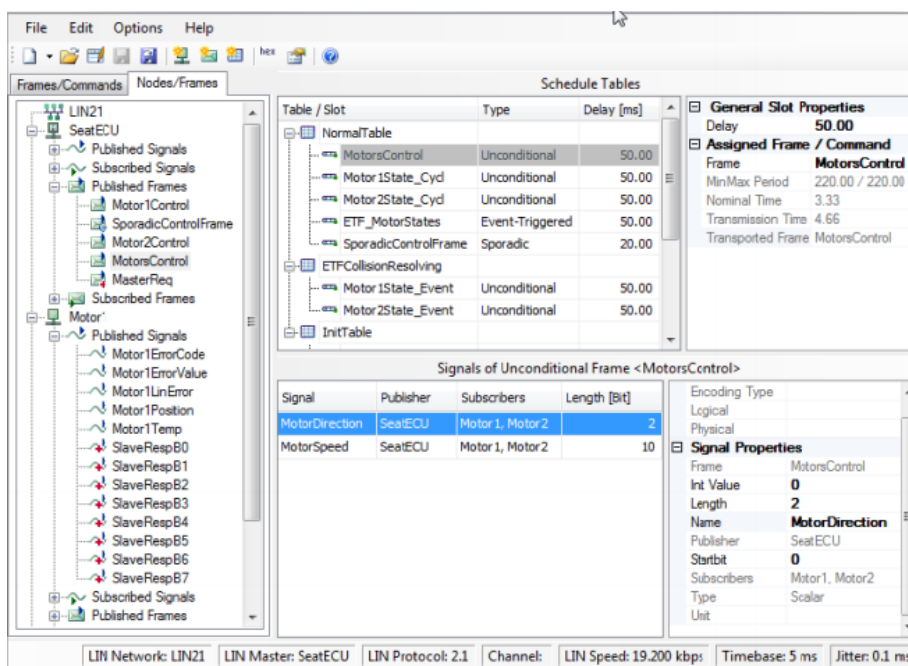
Jedná se o softwarový doplněk k programu Vector CANoe nezbytný pro práci se sběrnici LIN. Rozšíření podporuje vývoj, analýzu a testování této sběrnice podle specifikací LIN1.x, LIN2.x, SAE-J2602 a Cooling-Bus. [21]

CANoe.LIN nabízí možnost simulace jednotlivých uzlů sběrnice, včetně multi kanálových LIN master uzlů a spojení mezi nimi. Naměřené hodnoty je dále možno graficky zpracovávat do grafů, nebo je zprávu možné analyzovat z časového hlediska, to znamená rozdělit zprávu a určit čas po který je posílána hlavička, odpověď a jak dlouho trvá celková doba přenosu. Dále se dá odečíst celková rychlost přenosu, délka synchronizační pauzy, a mezirámcové mezery, anebo doba nečinnosti na sběrnici.



Obrázek 1.39: Trace window zpráv odposlechnutých na LIN sběrnici

Důležitou součástí rozšíření CANoe.LIN je takzvaný LDF explorer, který slouží pro vytváření, upravování a prohlížení jednotlivých zpráv a tím zpřehledňuje komunikaci mezi řídicí jednotkou a akčními členy a vytváří konfiguraci, ze které se dále vychází při následném vývoji určitého projektu.

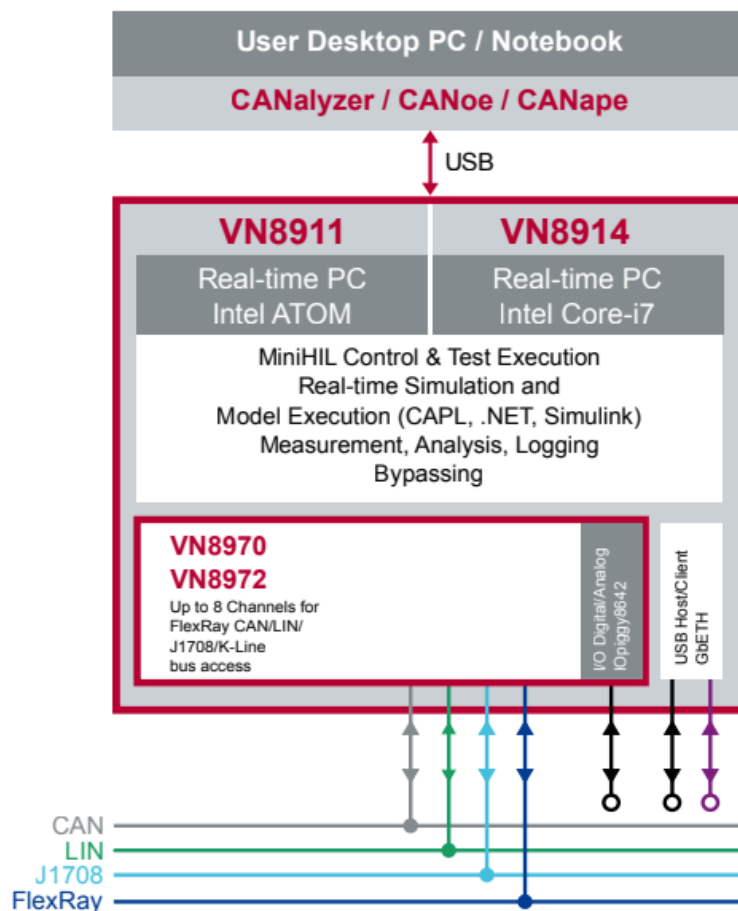


Obrázek 1.40: Ukázka rozšíření LDF explorer[22]

### 3.4.2 Převodník VN8914A

Převodník VN8914A spadá do kategorie hardwarových rozšíření ze série VN8900 s integrovaným procesorem pro zprostředkování komunikace počítače s vybranou automobilovou sběrnici. Analýzu a záznam signálů na sběrnici je schopen zaznamenávat v reálném čase. Nabízí

možnost připojení až 4 zařízení a pro komunikaci s počítačem lze využít klasické USB nebo ethernetové připojení, které lze použít i pro komunikaci se zvolenou sběrnici. [23]



Obrázek 1.41: *Blokové schéma zapojení převodníku [24]*

## 4 Ověření funkčnosti testeru

Ověření testovacího stanoviště bylo provedeno v laboratorních prostorách školy a dělí se na několik samostatných testů. První v pořadí byl proveden test pro správnou funkčnost testeru, který zahrnoval ověření bezchybné komunikace po sběrnici LIN. Tato komunikace byla analyzována s využitím softwarového prostředí CANoe od firmy Vector s využitím převodníku VN8914A a konfigurací určenou k analýze této sběrnice. Dále byla ověřena funkčnost aplikačního software v jednotlivých režimech pro oba motory. Zde bylo hlavním cílem odladit program a zajistit jeho funkčnost ve všech možných případech, které při jeho používání mohou nastat. V poslední řadě byla ověřena odolnost napájecího zdroje proti zkratu a byla změřena jeho zatěžovací charakteristika.

### 4.1 Ověření LIN komunikace

Pomocí programu CANoe byla analyzována komunikace mezi řídicím mikrokontrolerem a řadičem krokového motoru. Jedná se o záznam režimu autotest, kdy dochází k inicializaci měřeného motoru a následná kontrola nastavovaných pozic jako jsou maximum, minimum a nominální poloha.

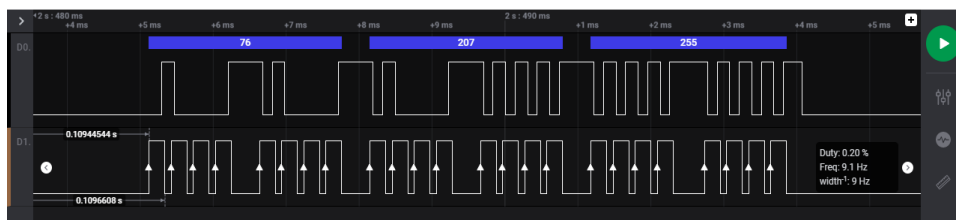
Frame Name	Id	Data	Publisher
starting up in wake mode			
MasterReq	3C (PID: 3C)	80 91 08 84 04 28 E4 10	HCM (real)
MasterReq	3C (PID: 3C)	80 89 08 F0 AB 25 00 E3	HCM (real)
MasterReq	3C (PID: 3C)	80 81 88 FF FF FF FF FF	HCM (real)
SlaveResp	3D (PID: 7D)	88 F0 AB 05 00 12 08 FF	Unknown node (real)
SlaveResp	3D (PID: 7D)	88 20 00 20 00 00 09 00	Unknown node (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 20 00 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 21 90 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 23 F8 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 26 60 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 28 C8 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 2B 30 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 2C 00 00	DHL (real)
AMIS_Get_Act_Pos_DHL	08 (PID: 08)	2C 00	
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 2C 00 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 2A 70 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 28 08 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 25 A0 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 23 38 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 20 D0 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 1E 68 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 1B F8 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 19 90 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 17 28 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 14 C0 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 14 00 00	DHL (real)
AMIS_Get_Act_Pos_DHL	08 (PID: 08)	20 00	
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 14 00 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 15 90 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 18 00 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 1A 68 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 1C D0 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 1F 38 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 20 00 00	DHL (real)
entering sleep mode due to bus idle timeout			

Obrázek 1.42: Záznam komunikace mezi krokovým motorem a řídicí jednotkou

### 4.2 Ověření měření polohy motorku

Pro další práci s přichozími daty z posuvného měřítka bylo nejprve potřeba analyzovat komunikaci na jeho datovém a hodinovém výstupu pomocí logického analyzátoru. Za předpokladu že data z posuvky přichází vždy když dojde ke změně polohy měřicí části byl

zaznamenán průběh při této změně. Díky tomuto průběhu bylo možné nastavit softwarově sběr dat a ten dále použít při měření a kontrole zvoleného elektrického motoru.



Obrázek 1.43: Data z posuvného měřítka SDA (horní průběh) a SCL (dolní průběh)

Kontrolní měření posuvného měřítka proběhlo se stejnosměrným motorem v režimu manual, kdy bylo možné ručně nastavit míru vysunutí hřídele motoru, se kterou se zároveň hýbalo měřítko. Díky manuálnímu nastavení vzdálenosti bylo možné odečítat data zobrazená na displeji měřítka a zároveň i data zobrazená na sériové lince pomocí sériového monitoru v programovacím prostředí arduino ide. Naměřená data, odečtená ze sériové linky jsou zobrazena v tabulce Tabulka 1.12: Data naměřená z posuvného měřítka začínaly v tomto momentě v nule, protože ale však není podstatné, z které pozice začne posuvné měřítko posílat hodnoty o vzdálenosti, ale záleží, jakou vzdálenost urazila hřídelka při testování může se stát, že naměřené hodnoty by se v následujícím měření od sebe lišily. Co však zůstává vždy stejné je úroveň PWM signálu na řídící elektrodě stejnosměrného motoru, kdy počáteční hodnota bude vždy 20 a koncová 230 a to z důvodu že Arduino má na výstupu osmibitový D/A převodník čili maximální možné číslo, které lze na výstupu nastavit je 255 a minimální 0. Tyto čísla však byly omezeny na zmiňované hodnoty z důvodu oblasti nehybnosti motorku, která vychází z grafu viz. Obrázek 1.15: .

Tabulka 1.12: Hodnoty měření pro kontrolu posuvného měřítka

čas	Data z posuvného měřítka (mm)	Hodnota PWM signálu na řídící elektrodě DC motoru
11:23:51	0,00	20
11:23:56	0,16	50
11:23:57	0,96	100
11:23:58	1,84	150
11:23:59	2,80	200
11:24:00.	3,19	230

### 4.3 Ověření testovací funkce Auto

Testovací režim autotest byl odzkoušen a naměřené hodnoty byly odečteny ze sériového portu v prostředí Arduino IDE a u krokového motoru i v programu CANoe. Pro oba typy elektrických motorů bylo provedeno měření s funkčním i nefunkčním motorem. Výsledek z měření pro stejnosměrný motor je vidět v tabulce Tabulka 1.13: a Tabulka 1.14: výsledek testů pro měření krokového motoru pak v Tabulka 1.15: a Tabulka 1.16:.

## Ověření funkčnosti testeru

Tabulka 1.13: *Odečtené data ze sériového portu režimu autotest DC – motor OK*

čas	Data z posuvného měřítka (mm)	Rozdíl naměřených dat a naměřených dat z předchozí polohy
11:29:00	-0,02	-
11:29:05	3,17	3,19
11:29:09	1,63	1,56
11:29:09	Výsledek měření: DC Motor OK	

Protože byl rozdíl hodnot v obou polohách v rozmezí tolerance definované pro správně fungující elektrický motor, Vyhodnocení funkčnosti motoru mělo výsledek "motor OK".

Tabulka 1.14: *Odečtené data ze sériového portu režimu autotest DC – motor NOK*

čas	Data z posuvného měřítka (mm)	Rozdíl naměřených dat a naměřených dat z předchozí polohy
11:29:43	8,51	-
11:29:48	8,51	0
11:29:52	8,51	0
11:29:52	Výsledek měření: DC Motor NOK	

Posuvné měřítko v tomto případě nevykonalo žádný pohyb, a proto rozdíl naměřených hodnot byl vždy nulový a mimo nastavené meze tolerance. Z tohoto důvodu vyhodnocení měření funkčnosti motoru mělo výsledek "motor NOK".

Měření krokového motoru probíhalo obdobně s tím rozdílem, že na sériové lince byl vypsán i datový rámec zprávy GetActPos s aktuální pozicí a každý autotest zde začíná odpovědí na inicializační zprávu zprávou GetFullStatus. Průběh ze sériové linky je totožný s průběhem zobrazeným viz. Obrázek 1.42: , proto jsou v této tabulce uvedeny pouze hodnoty z posuvného měřítka bez zpráv ze sběrnice LIN.

Tabulka 1.15: *Odečtené data ze sériového portu režimu autotest KM – motor OK*

čas	Data z posuvného měřítka (mm)	Rozdíl naměřených dat a naměřených dat z předchozí polohy
21:09:46	5,76	-
21:09:48	-6,16	11,92
21:09:50	0	6,16
21:09:50	Výsledek měření: KM Motor OK	

Tabulka 1.16: *Odečtené data ze sériového portu režimu autotest KM – motor NOK*

čas	Data z posuvného měřítka (mm)	Rozdíl naměřených dat a naměřených dat z předchozí polohy
21:09:59	5,76	-
21:10:01	5,76	0
21:10:03	5,76	0
21:10:03	Výsledek měření: KM Motor OK	

#### 4.4 Ověření testovací funkce Manual

Režim změny polohy hřídele motoru manuálně byl proveden již v tabulce Tabulka 1.12: kdy bylo spolu s tímto režimem u DC motoru rovněž otestováno i posuvné měřítko. U krokového motoru probíhalo ověření obdobně opět se zobrazenou komunikací po sběrnici LIN. Obrázek 1.44: ukazuje záznam tohoto měření, kdy opět na začátku proběhla inicializace a poté už motor čekal na zadanou polohu určenou ve zprávě s identifikátorem "08". Této polohy motorek dosáhl vždy téměř okamžitě jak je vidět z jeho odpovědi.

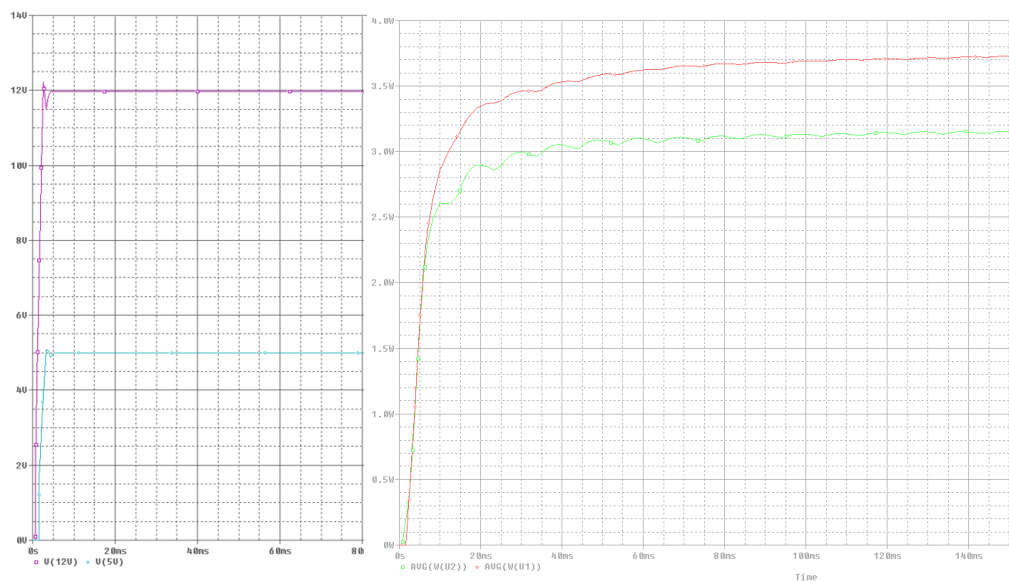
Frame Name	Id	Data	Publisher
waking up due to external wakeup frame or bus traffic			
MasterReq	3C (PID: 3C)	80 91 08 84 04 28 E4 10	HCM (real)
MasterReq	3C (PID: 3C)	80 89 08 F0 A8 25 00 E3	HCM (real)
MasterReq	3C (PID: 3C)	80 81 88 FF FF FF FF FF	HCM (real)
SlaveResp	3D (PID: 7D)	88 F0 A8 05 00 12 08 FF	Unknown node (real)
SlaveResp	3D (PID: 7D)	88 20 00 20 00 00 09 00	Unknown node (real)
08 (PID: 08)	08 (PID: 08)	14 00	
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 20 00 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 1E 08 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 1B 98 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 19 30 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 16 C8 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 14 60 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 14 00 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 14 00 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 14 00 00	DHL (real)
08 (PID: 08)	08 (PID: 08)	15 00	
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 14 00 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 15 00 00	DHL (real)
08 (PID: 08)	08 (PID: 08)	15 80	
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 15 00 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 15 80 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 15 80 00	DHL (real)
AMIS_Get_Act_Pos_DHL	24 (PID: 64)	88 15 80 00	DHL (real)
08 (PID: 08)	08 (PID: 08)	15 00	

Obrázek 1.44: Komunikace mezi KM a řídicí jednotkou při režimu manual



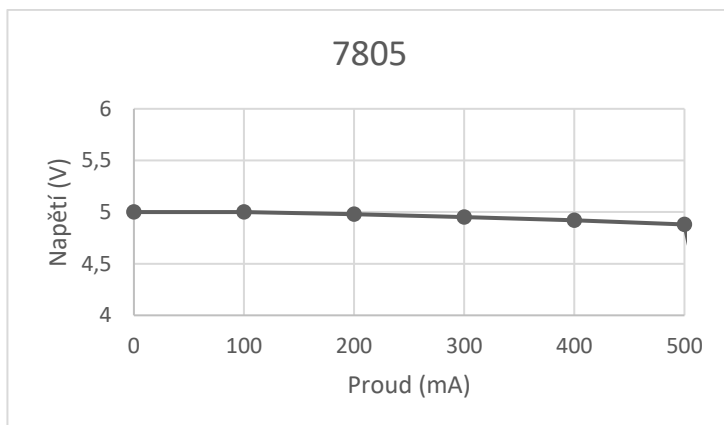
#### 4.5 Oživení a otestování napájecího zdroje

Samostatný test napájecího zdroje pro testovací stanoviště se dělí na dvě části. Nejprve byla provedena simulace zapojení zdroje v prostředí OrCad ještě před jeho zhotovením. Zde byly odsimulovány průběhy s maximálním proudovým odběrem. Z průběhů výstupních napětí lze vidět, že při maximálním požadovaném odebíraném proudu 0,5 A pro obě větve zdroje nemají stabilizátory problém udržovat hladinu výstupního napětí na stanovené hodnotě 5 V a 12 V. Jejich ztrátový výkon byl v simulaci 3 W pro 12 V větev a 3,5 W pro 5 V větev viz. Obrázek 1.45: .

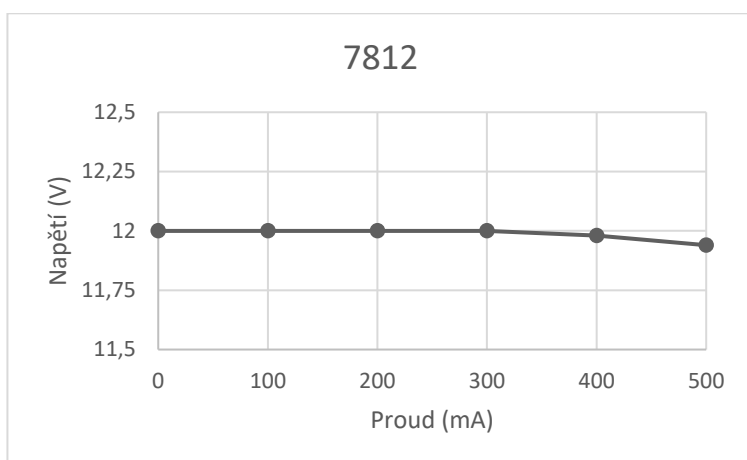


Obrázek 1.45: Zleva: průběh výstupního napětí ze stabilizátoru 7812 (fialový) 7805 (modrý) a průběh výkonové ztráty na stabilizátorech 7812 (zelený) a 7805 (červený)

Na zhotoveném zdroji byla změřena zatěžovací charakteristika s elektronickou zátěží EA-EL9080-170B. Zde se kontroloval pokles výstupního napětí podle nastavovaného proudu odebíraného zátěží od stavu na prázdko až po maximální požadovaný odběr a to 0,5 A. Z naměřených hodnot se určil vnitřní odpor zdroje, který pro napětřovou větev se stabilizátorem 7805 podle Ohmova zákona vyšel  $R_i = 0,3 \Omega$  a pro napětřovou větev se stabilizátorem 7812  $R_i = 0,15 \Omega$ . Zdroj se při napájení veškeré řídicí elektroniky po dobu testování nepřehříval a nedošlo k žádným poruchám, svému účelu dostačuje.



Obrázek 1.46: *Závislost výstupního napětí stabilizátoru 7805 na proudu protékajícím elektronickou zátěží*



Obrázek 1.47: *Závislost výstupního napětí stabilizátoru 7812 na proudu protékajícím elektronickou zátěží*

## Závěr

V diplomové práci byly popsány komunikační prostředky automobilů používané v současné době pro osvětlovací automobilové systémy, konkrétněji byla popsána vnitřní struktura řídicí elektroniky moderního světlometu a následně také automobilová sběrnice LIN. Informace o sběrnici se vztahují ke způsobu komunikace po sběrnici, jak je realizována její hardwarová vrstva a jak vypadá vzorový rámec zprávy, který se po sběrnici posílá. V závěru teoretické části je stručný popis krokového a stejnosměrného motoru v praxi používaného pro nastavování úrovně osvětlení vozovky v horizontálním a vertikálním směru.

V praktické části této práce byl navržen napájecí zdroj pro testovací přípravek určený k analýze elektrických motorů. Navrženo bylo elektrické schéma zdroje i jeho 3D model, díky kterému bylo možné určit jaké rozměry budou potřeba při dalším návrhu celkové konstrukce testovacího boxu a jak bude zdroj chlazen.

Popis testovacího přípravku se dále dělí na část pro analýzu krokového motoru a část pro analýzu stejnosměrného motoru. Každá z těchto částí byla navržena samostatně a byly popsány jednotlivé použité komponenty jako například řídicí jednotka, měřicí člen, nebo zobrazovací jednotka. Řízení obou částí obstarává mikrokontrolér na bázi vývojové platformy Arduino, a obě měřicí části dohromady tvoří jedno testovací stanoviště.

Kapitoly zaměřené na realizaci testeru se zvlášť zabývají analýzou krokového a stejnosměrného motoru. Je zde popsána finální podoba testovacího stanoviště a jsou zde zobrazeny 3D modely jednotlivých komponent. V kapitole realizace napájecího zdroje je zobrazeno rozložení elektrických součástek na desce plošného spoje a je zde také jeho finální podoba. Dále je popsán princip řídicího programu, který se dá rozdělit na tři podprogramy. Program pro výběr testovaného motoru, dále testovací program pro automatické nastavení a seřízení motoru a režim manuál, který dovoluje manuálně nastavit motor do libovolné polohy.

V předposlední kapitole realizace se nachází rozpis jednotlivých zpráv posílaných po sběrnici LIN, kterými se řídí krokový motor. Všechny zprávy jsou zapsány v přehledných tabulkách, které jsou doplněny krátkým popisem. Závěr kapitoly realizace je věnován programu CANoe Vector, který je použit pro analýzu komunikace sběrnice LIN. V poslední části práce je popsáno ověření funkčnosti testeru a rozpis jednotlivých dílčích testů s doloženými průběhy získanými při testování.

Cílem práce bylo vyvinout testovací prostředek pro analýzu vybraných typů elektrických motorů. Požadavky na tuto práci byly splněny ve stanoveném termínu a ve všech bodech, a navíc byl k testovacímu stanovišti navržen a vyroben napájecí zdroj, který byl zohledněn při konstrukčním řešení testeru. Testovací box bude potřeba ještě při reálném provozu ve firmě řádně odzkoušet, testům v laboratorních podmínkách školy ale vyhověl. Jedná se o prototypový návrh, rozdělený do jednotlivých modulů, do budoucna by se všechny tyto moduly mohly umístit na jednotný shield pro větší robustnost a spolehlivost přípravku.



## Použitá literatura

- [1] VLK, František. Elektronické systémy motorových vozidel 2. Brno: TINLIB, 2002. ISBN 80-238-7282-6.
- [2] OSVĚTLENÍ MOTOROVÝCH VOZIDEL. Sinz [online]. Brno, 2006 [cit. 2020-05-12]. Dostupné z: <http://www.sinz.cz/archiv/docs/si-2006-05-292-300.pdf>
- [3] HEADLAMP LEVELLING SYSTEM. In: Hella [online]. [cit. 2020-05-12]. Dostupné z: <https://www.hella.com/techworld/uk/Technical/Automotive-lighting/Headlamp-levelling-system-838/>
- [4] LIN BUS EXPLAINED. Csselectronics [online]. [cit. 2020-02-21]. Dostupné z: <https://www.csselectronics.com/screen/page/lin-bus-protocol-intro-basics/language/en>
- [5] LIN - Local Interconnect Network. Automatizace [online]. 2004 [cit. 2020-02-21]. Dostupné z: <https://automatizace.hw.cz/clanek/2005101501>
- [6] SUTORÝ, Tomáš [online]. Local Interconnect Network. [cit. 2019-11-8]. Dostupné z <http://www.elektrorevue.cz/clanky/04012/index.html>
- [7] Pavel Brandštetter, T. P. (2007). ELEKTRONIKA Prvky elektronických obvodů. Ostrava.
- [8] ARDUINO MEGA 2560 REV3. Arduino [online]. [cit. 2020-05-12]. Dostupné z: <https://store.arduino.cc/arduino-mega-2560-rev3>
- [9] KOUŘIL, Daniel. Stabilizátory a zdroje napětí KZPE. Ostrava, 2019. Pomocné výukové materiály. VŠB TUO.
- [10] KOUŘIL, Daniel. Výpočty ztrátového výkonu a návrh chlazení KZPE. Ostrava, 2019. Pomocné výukové materiály. VŠB TUO.
- [11] Chladič D01. In: TME [online]. [cit. 2020-05-12]. Dostupné z: <https://www.tme.eu/cz/details/d01/chladice/>
- [12] 100% kompatibilní klon Arduino MEGA 2560 R3. In: GME [online]. [cit. 2020-05-12]. Dostupné z: <https://www.gme.cz/100-kompatibilni-klon-arduino-mega-2560-r3-usb-b-atmega16u2>
- [13] TJA1028 LIN transceiver with integrated voltage regulator. Nxp [online]. 2012 [cit. 2020-03-02]. Dostupné z: <https://www.nxp.com/docs/en/data-sheet/TJA1028.pdf>
- [14] ACS712 Datasheet. Alldatasheet [online]. [cit. 2020-05-12]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/174116/ALLEGRO/ACS712.html>
- [15] Arduino proudový senzor ACS712. In: Arduino-shop [online]. [cit. 2020-05-12]. Dostupné z: [https://arduino-shop.cz/arduino/1072-arduino-5a-proudovy-senzor-acs712.html?gclid=CjwKCAjwkun1BRAIEiwA2mJRWa3KoSobEnULFEh313MA-T2vzqdcNfYbq3eo9j-OO7ayc1\\_WTTm1lRoCo88QAvD\\_BwE](https://arduino-shop.cz/arduino/1072-arduino-5a-proudovy-senzor-acs712.html?gclid=CjwKCAjwkun1BRAIEiwA2mJRWa3KoSobEnULFEh313MA-T2vzqdcNfYbq3eo9j-OO7ayc1_WTTm1lRoCo88QAvD_BwE)

- [16] Posuvné měřítko s displejem. In: Hadex [online]. [cit. 2020-05-12]. Dostupné z: <https://www.hadex.cz/r296-posuvne-meritko---suplera-150mm-digitalni-presnost-01mm/>
- [17] I2C LCD displej 16x2. In: Hwkitchen [online]. [cit. 2020-05-12]. Dostupné z: <https://www.hwkitchen.cz/iic-i2c-lcd-displej-16x2-modry-s-podsvetlenim/>
- [18] Lcd-seriove-rozhrani. In: Arduino-shop [online]. [cit. 2020-05-12]. Dostupné z: <https://arduino-shop.cz/arduino/1071-lcd-seriove-rozhrani-iic-i2c-twi-sp-1602.html>
- [19] AMIS-30621 LIN Microstepping Motordriver. Alldatasheet [online]. [cit. 2020-05-01]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/192177/AMI/AMIS-30621.html>
- [20] Product Information CANoe. Vector [online]. [cit. 2020-05-03]. Dostupné z: [https://assets.vector.com/cms/content/products/canoe/canoe/docs/Product%20Informations/CANoe\\_ProductInformation\\_EN.pdf](https://assets.vector.com/cms/content/products/canoe/canoe/docs/Product%20Informations/CANoe_ProductInformation_EN.pdf)
- [21] Product Information CANoe.LIN. Vector [online]. [cit. 2020-05-04]. Dostupné z: [https://assets.vector.com/cms/content/products/canoe/\\_lin/CANoe\\_LIN\\_ProductInformation\\_EN.pdf](https://assets.vector.com/cms/content/products/canoe/_lin/CANoe_LIN_ProductInformation_EN.pdf)
- [22] LDF Explorer for viewing and editing LIN Description Files. In: Vector [online]. [cit. 2020-05-06]. Dostupné z: [https://assets.vector.com/cms/content/products/canoe/\\_lin/CANoe\\_LIN\\_ProductInformation\\_EN.pdf](https://assets.vector.com/cms/content/products/canoe/_lin/CANoe_LIN_ProductInformation_EN.pdf)
- [23] VN8900 Interface Family Manual. Vector [online]. [cit. 2020-05-06]. Dostupné z: [https://assets.vector.com/cms/content/products/VN89xx/docs/VN89xx\\_Manual\\_EN.pdf](https://assets.vector.com/cms/content/products/VN89xx/docs/VN89xx_Manual_EN.pdf)
- [24] Operator side (user PC) and measurement side. In: Vector [online]. [cit. 2020-05-06]. Dostupné z: [https://assets.vector.com/cms/content/products/VN89xx/docs/VN89xx\\_Manual\\_EN.pdf](https://assets.vector.com/cms/content/products/VN89xx/docs/VN89xx_Manual_EN.pdf)

---

---